**Лабораторная работа №2**

*Тема: анализ данных.*

**Цель работы:** Получение практических навыков анализа данных.

**Задание:** Используя программное средство Weka, выполните анализ данных согласно полученному варианту. Работа состоит из нескольких этапов:

1. Подготовка данных для анализа в полученной согласно варианту предметной области (атрибутов должно быть не менее 10, строк с данными не менее 100, строки должны быть уникальными)
2. Загрузка данных в систему, рассмотрение описания данных (максимальных, минимальных значений и т.д.)
3. Построение моделей различными методами:

* Регрессионной,
* Классификации
* Кластеризации
* Ассоциативной

1. Исследование моделей, их интерпретация и выводы о возможности их применения

**Отчет** по лабораторной работе должен содержать:

1. Фамилию и номер группы, задание
2. Описание данных
3. Описание процесса построения моделей
4. Описание полученного результата (с визуализацией)
5. Интерпретация полученных результатов и выводы
6. Листинги данных и моделей.

# Основные теоретические сведения

Weka (Waikato Environment for Knowledge Analysis) — свободное программное обеспечение для анализа данных, написанное на Java в университете Уайкато (Новая Зеландия), распространяющееся по лицензии GNU GPL. Система представляет собой систему библиотек функции обработки данных, плюс несколько графических интерфейсов к этим библиотекам. Основной интерфейс системы - Explorer. Он позволяет выполнять практически все действия, которые предусмотрены в системе.

Также в системе Weka предусмотрены другие интерфейсы - Knowledge Flow для работы с большими массивами данных (Explorer загружает все данные в память сразу, и потому работа с большими массивами затруднена) и Experimenter для экспериментального подбора наилучшего метода анализа данных.

*Weka* предоставляет доступ к SQL-базам через Java Database Connectivity (JDBC) и в качестве исходных данных может принимать результат SQL-запроса. Возможность обработки множества связанных таблиц не поддерживается, но существуют утилиты для преобразования таких данных в одну таблицу, которую можно загрузить в *Weka.*

Explorer имеет несколько панелей.

1. Панель предобработки Preprocess panel позволяет импортировать данные из базы, CSV файла и т. д., и применять к ним алгоритмы фильтрации, например, переводить количественные признаки в дискретные, удалять объекты и признаки по заданному критерию.
2. Панель классификации Classify panel позволяет применять алгоритмы классификации и регрессии (в Weka они не различаются и называются classifiers) к выборке данных, оценивать предсказательную способность алгоритмов, визуализировать ошибочные предсказания, ROC-кривые, и сам алгоритм, если это возможно (в частности, решающие деревья).
3. Панель поиска ассоциативных правил Associate panel решает задачу выявления всех значимых взаимосвязей между признаками.
4. Панель кластеризации Cluster panel даёт доступ к алгоритму k-средних, EM-алгоритму для смеси гауссианов и другим.
5. Панель отбора признаков Select attributes panel даёт доступ к методам отбора признаков.
6. Панель визуализации Visualize строит матрицу графиков разброса (scatter plot matrix), позволяет выбирать и увеличивать графики, и т. д.

WEKA использует Java, так что если на вашем компьютере нет JRE, выберите для установки версию WEKA, включающую в себя JRE.

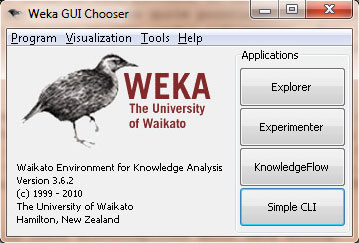


Рисунок 1 - Стартовое окно WEKA

При запуске WEKA, пакет предлагает вам на выбор 4 графических интерфейса для работы с WEKA и вашими данными. Выберете Explorer.

Анализ данных подразумевает наличие самих данных в системе. Для того чтобы загрузить данные в WEKA, их следует преобразовать в формат, понятный этому программному пакету. Наиболее подходящим форматом для загрузки данных в WEKA является формат Attribute-Relation File Format (ARFF), который сначала определяет тип загружаемых данных, а потом указывает собственно данные.

В файле формата ARFF вы указываете название и тип данных для каждого столбца таблицы, а затем собственно данные по строкам.

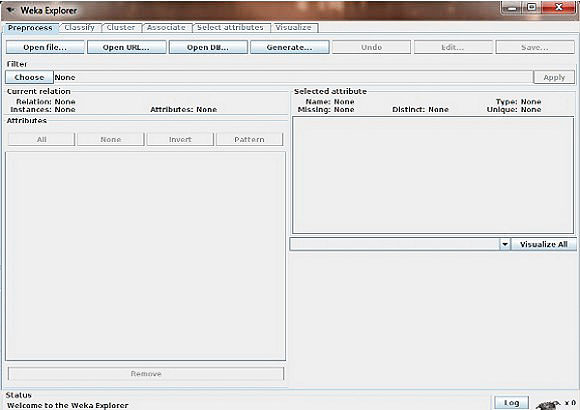


Рисунок 2 - Окно WEKA Explorer

Подготовка данных состоит в добавлении полей метаданных в начало файла. На отдельных строчках добавляется следующая информация:

-названия зависимости @relation имя,

-описания атрибутов @attribute: имя, тип

-@data перед началом самих данных.

Различают следующие типы данных:

-численные (numeric, real, integer),

-перечислимые(nominal) (задаются перечислением вида {i1, ..., in}),

-строковые (string),

-дата (date [date format]).

- составной тип (relational),

## Пример использования различных моделей анализа данных в Weka

### Регрессия

В моделях регрессионного анализа используются всего два типа данных: NUMERIC и DATE.

Метод регрессионного анализа является самым простым и, пожалуй, наименее эффективным методом интеллектуального анализа данных (удивительно, как часто эти качества сопутствуют друг другу). Самая простая модель анализа использует один входной (независимый) параметр и один результирующий (зависимый) параметр, модель можно усложнить, добавив несколько десятков входных параметров, но в любом случае общий подход будет один и тот же: на основании нескольких независимых переменных определяется один зависимый результат. Таким образом, модель регрессионного анализа используется для прогнозирования значения одной зависимой переменной, исходя из известных значений нескольких независимых параметров.

Наверняка, каждый из нас хотя бы раз сталкивался с регрессионной моделью, а может быть, и проводил в уме самостоятельный регрессионный анализ. Наиболее очевидный пример – определение стоимости дома. Цена на дом (зависимая переменная) определяется несколькими независимыми параметрами: какова площадь дома и размер участка, используется ли в оформлении кухни гранитные плиты, каково качество и срок службы сантехники и так далее. Так что, если вам случалось когда-либо продавать или покупать дом, то, скорее всего, вы использовали регрессионный анализ для определения его цены. Вы оценивали параметры похожих домов в этом же районе и цену, по которой эти дома были проданы (т.е. создавали модель), а затем подставляли параметры вашего дома в полученную зависимость и рассчитывали предполагаемую стоимость вашего дома.

Давайте воспользуемся моделью регрессионного анализа для определения цены дома и разберем конкретный пример. В таблице внизу указаны фактические параметры домов, выставленных на продажу в моем районе. На основании этих данных я попробую оценить стоимость моего дома (и воспользуюсь этими результатами, чтобы опротестовать предъявленную мне сумму налога на недвижимость).

Таблица 1. Регрессионная модель оценки стоимости дома

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Площадь дома (кв.футы)** | **Размер участка** | **Количество спален** | **Гранитная отделка на кухне** | **Современное сантехническое оборудование?** | **Продажная цена** |
| 3529 | 9191 | 6 | 0 | 0 | $205,000 |
| 3247 | 10061 | 5 | 1 | 1 | $224,900 |
| 4032 | 10150 | 5 | 0 | 1 | $197,900 |
| 2397 | 14156 | 4 | 1 | 0 | $189,900 |
| 2200 | 9600 | 4 | 0 | 1` | $195,000 |
| 3536 | 19994 | 6 | 1 | 1 | $325,000 |
| 2983 | 9365 | 5 | 0 | 1 | $230,000 |
|  |  |  |  |  |  |
| 3198 | 9669 | 5 | 1 | 1 | ???? |

Файл с данными (отношение \ таблица Дом, атрибуты размер дома, ванная и т.д. и собственно данные через запятую):

@RELATION house

@ATTRIBUTE houseSize NUMERIC

@ATTRIBUTE lotSize NUMERIC

@ATTRIBUTE bedrooms NUMERIC

@ATTRIBUTE granite NUMERIC

@ATTRIBUTE bathroom NUMERIC

@ATTRIBUTE sellingPrice NUMERIC

@DATA

3529,9191,6,0,0,205000

3247,10061,5,1,1,224900

4032,10150,5,0,1,197900

2397,14156,4,1,0,189900

2200,9600,4,0,1,195000

3536,19994,6,1,1,325000

2983,9365,5,0,1,230000

 Запустите WEKA и выберите опцию Explorer. В результате откроется закладка Preprocess окна Explorer. Щелкните на кнопке Open File и выберите созданный вами ARFF-файл. Окно WEKA Explorer с загруженными данными о домах показано на рисунке 3.

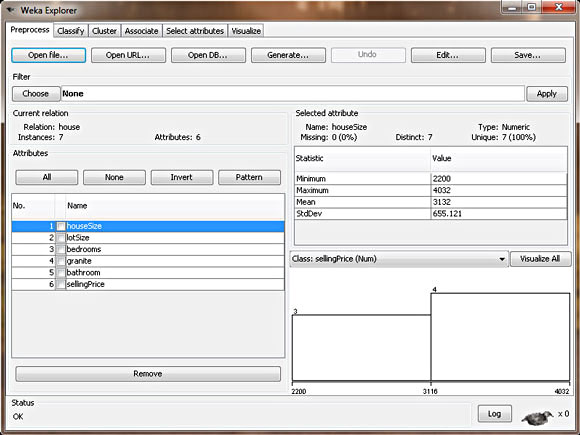


Рисунок 3- Окно WEKA Explorer с загруженными данными о домах

В этом окне вы можете проверить данные, на основании которых вы собираетесь строить модель. В левой части окна Explorer показаны параметры объектов (Attributes), которые соответствуют заголовкам столбцов нашей исходной таблицы, а также указано количество объектов (Instances), т.е. строк таблицы. Если вы щелкните мышкой на одном из заголовков столбцов, то в правой панели будет выведена полная информация о наборе данных в данном столбце. Например, если мы выберем столбец houseSize в левой панели (он выбран по умолчанию), то в правой панели отобразится дополнительная статистическая информация по этому столбцу. Будет показано максимальное значение в столбце (4032 кв.фута) и минимальное значение (2200 кв.футов). Кроме того, будет подсчитано среднее значение (3131 кв.фут) и стандартное отклонение (655 кв.футов) (стандартное отклонение – статистический показатель рассеивания значений случайной величины). Наконец, здесь же вам предлагается возможность визуального анализа данных (кнопка Visualize All). Поскольку в нашей таблице данных не так много, то их визуальное отображение не дает такой наглядной аналитической картины, как в случае использования сотен или тысяч показателей.

Для того чтобы создать модель, откройте закладку **Classify**. В качестве первого шага, нам надо выбрать тип модели для анализа, чтобы указать WEKA, каким образом мы хотим анализировать наши данные, и какую модель построить:

1. Щелкните на копке **Choose** и разверните меню **functions**.
2. Выберите опцию **LinearRegression**.

Таким образом, мы указали WEKA, что мы хотим создать модель регрессионного анализа. Как вы заметили, меню включает целое множество моделей. Множество! Это еще раз подчеркивает факт нашего весьма поверхностного знакомства с областью интеллектуального анализа данных. Обратите внимание: в меню включена опция **SimpleLinearRegression**, однако мы не используем ее, поскольку этот тип модели определяет значение зависимой переменной по значениям одного независимого параметра, а у нас их целых шесть. Если вы выбрали правильную модель, то окно WEKA Explorer должно выглядеть так, как показано на рисунке 4.

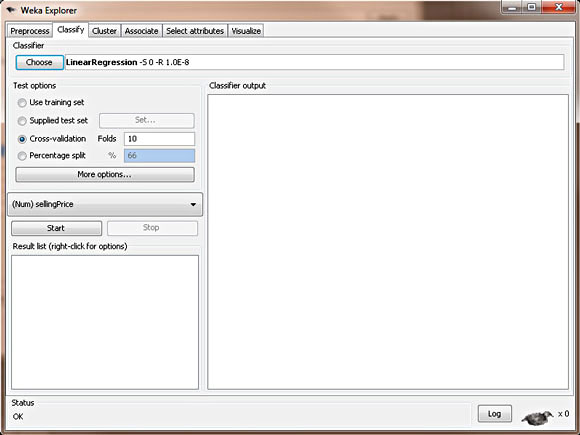


Рисунок 4 - Модель линейного регрессионного анализа WEKA

После того, как мы выбрали тип модели, нужно указать WEKA, какие данные должны использоваться для ее создания. Несмотря на то, что ответ на этот вопрос для нас вполне очевиден – нужно взять данные из созданного нами ARFF-файла – существует несколько других, более сложных, возможностей предоставления данных для анализа. Опция **Supplied test set** позволяет указать дополнительный набор тестовых данных для модели, опция **Cross-validation** использует несколько наборов данных, усредняет их и строит модель на основе средних значений, а опция**Percentage split** использует в качестве базы для модели процентили набора данных. Эти способы применяются для создания аналитических моделей, которые мы рассмотрим в следующих статьях этой серии. В случае регрессионного анализа нам нужна опция **Use training set**. В этом случае WEKA создаст модель на базе данных из загруженного ARFF-файла.

Завершающий этап создания модели – выбор зависимой переменной (столбца, в котором находится неизвестное нам значение, которое требуется рассчитать). В нашем примере – это цена дома, так как именно это значение мы и хотим узнать. Сразу после секции Test options находится раскрывающийся список, в котором вам нужно выбрать зависимый параметр. По умолчанию должен быть выбран атрибут **sellingPrice**. Если это не так, выберите сами этот параметр.

Мы определили все параметры и можем приступить к созданию модели. Нажмите кнопку **Start**. В результате окно WEKA должно выглядеть так, как показано на рисунке 5.

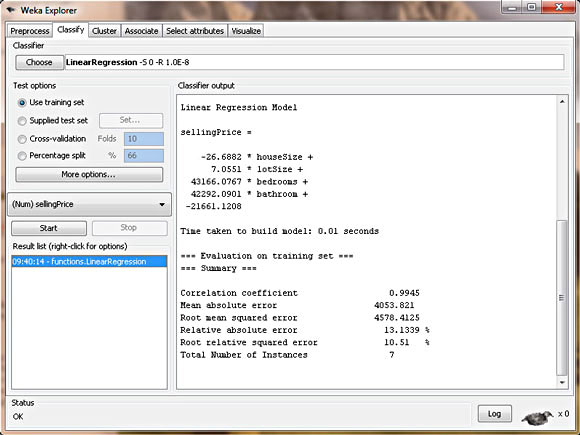


Рисунок 5 - Регрессионная модель WEKA для расчета стоимости дома

Готовая модель регрессионного анализа

sellingPrice = (-26.6882 \* houseSize) +

(7.0551 \* lotSize) +

(43166.0767 \* bedrooms) +

(42292.0901 \* bathroom)

- 21661.1208

Рассчитаем цену конкретного дома:

sellingPrice = (-26.6882 \* 3198) +

(7.0551 \* 9669) +

(43166.0767 \* 5) +

(42292.0901 \* 1)

- 21661.1208

sellingPrice = 219,328

Рассмотрим зависимости между данными нашей модели и постараемся сделать определенные выводы относительно правил формирования цен на недвижимость.

* Гранитные элементы в оформлении кухни не влияют на цену дома — WEKA использует только те данные, которые, согласно статистике, влияют на точность модели (влияние каждого независимого параметра на зависимую переменную определяется с помощью коэффициента детерминации R-квадрат, обсуждение которого выходит за рамки этой статьи). Таким образом, параметры, не имеющие достаточного влияния на зависимую переменную, в модели не учитываются. Наша регрессионная модель свидетельствует о том, что использование гранита на кухне не влияет на цену дома.
* Состояние ванных комнат и сантехники влияет на цену дома — поскольку мы используем значения 0 или 1 в качестве показателя модернизации ванных комнат, то соответствующий коэффициент регрессионной модели демонстрирует нам, как современное сантехническое оборудование влияет на цену дома, а именно добавляет 42292$ к его цене.
* Большая площадь дома снижает его цену — Согласно модели WEKA, по мере роста площади домов, цена снижается. Это следует из того, что модель включает переменную houseSize с отрицательным коэффициентом. Что же получается? Увеличение площади дома на 1 кв.фут снижает его стоимость на 26$? Подобное утверждение кажется очевидной бессмыслицей. Мы же рассматриваем дома в Америке: чем больше, тем лучше, особенно в Техасе, где я живу. Как же это понимать? Это классический пример случая «каков вопрос, таков и ответ». На самом деле, размер дома не является независимой величиной. Этот параметр связан, например, с количеством спален - очевидно, что в больших домах и количество спален больше. Так что наша модель, увы, не идеальна, но мы можем ее поправить. Запомните: закладка Preprocess позволяет удалить столбцы из набора данных. В качестве самостоятельного упражнения, удалите столбец houseSize и создайте новую модель. Проверьте, как изменение набора данных отразится на цене дома, и какая из двух моделей больше соответствует реальности (уточненная цена моего дома $217,894).

### Классификация

Метод классификации (также известный как метод классификационных деревьев или деревьев принятия решений) - это алгоритм анализа данных, который определяет пошаговый способ принятия решения в зависимости от значений конкретных параметров. Дерево этого метода имеет следующий вид: каждый узел представляет собой точку принятия решения на основании входных параметров. В зависимости от конкретного значения параметра вы переходите к следующему узлу, от него – к следующему узлу, и так далее, пока не дойдете до листа, который и дает вам окончательное решение. Звучит довольно запутанно, но на самом деле метод достаточно прямолинеен. Давайте обратимся к конкретному примеру.

[ Вы будете читать этот раздел? ]

/ \

Да Нет

/ \

[Вы его поймете?] [Не изучите]

/ \

Да Нет

/ \

[Изучите] [Не изучите]

Это простое классификационное дерево определяет ответ на вопрос «Изучите ли вы принцип построения классификационного дерева?» В каждом узле вы отвечаете на соответствующий вопрос и переходите по соответствующей ветке к следующему узлу, до тех пор, пока не дойдете до листа с ответом «да» или «нет». Эта модель применима к любым сущностям, и вы сможете ответить, в состоянии ли эти сущности изучить классификационные деревья, с помощью двух простых вопросов. В этом и состоит основное преимущество классификационных деревьев – они не требует чрезмерного количества информации для создания достаточно точного и информативного дерева решений.

#### J4.8 (модификация С4.5)

Набор данных, который мы будем использовать для примера классификационного анализа, содержит информацию, собранную дилерским центром BMW. Центр начинает рекламную компанию, предлагая расширенную двухгодичную гарантию своим постоянным клиентам. Подобные компании уже проводились, так что дилерский центр располагает 4500 показателями относительно предыдущих продаж с расширенной гарантией. Этот набор данных обладает следующими атрибутами:

* Распределение по доходам [0=$0-$30k, 1=$31k-$40k, 2=$41k-$60k, 3=$61k-$75k, 4=$76k-$100k, 5=$101k-$150k, 6=$151k-$500k, 7=$501k+]
* Год/месяц покупки первого автомобиля BMW
* Год/месяц покупки последнего автомобиля BMW
* Воспользовался ли клиент расширенной гарантией

Загрузите файл bmw-training.arff (см.раздел [Загрузка](http://www.ibm.com/developerworks/ru/library/os-weka2/index.html#download)) в программный пакет WEKA, используя те же шаги, которые мы проделали для загрузки данных в случае регрессионного анализа. Замечание: в предлагаемом файле содержатся 3000 из имеющихся 4500 записей. Мы разделили набор данных так, чтобы часть их использовалась для создания модели, а часть – для проверки ее точности, чтобы убедиться, что модель не является подогнанной под конкретный набор данных.

Откройте закладку Classify, выберите опцию trees, а затем опцию J48.Убедитесь, что выбрана опция Use training set, чтобы пакет WEKA при создании модели использовал именно те данные, которые мы только что загрузили в виде файла. Нажмите кнопку Start и предоставьте WEKA возможность поработать с нашими данными. Результирующая модель должна выглядеть так:

Number of Leaves : 28

Size of the tree : 43

Time taken to build model: 0.18 seconds

=== Evaluation on training set ===

=== Summary ===

Correctly Classified Instances 1774 59.1333 %

Incorrectly Classified Instances 1226 40.8667 %

Kappa statistic 0.1807

Mean absolute error 0.4773

Root mean squared error 0.4885

Relative absolute error 95.4768 %

Root relative squared error 97.7122 %

Total Number of Instances 3000

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class

0.662 0.481 0.587 0.662 0.622 0.616 1

0.519 0.338 0.597 0.519 0.555 0.616 0

Weighted Avg. 0.591 0.411 0.592 0.591 0.589 0.616

=== Confusion Matrix ===

a b <-- classified as

1009 516 | a = 1

1. 65 | b = 0

Наиболее существенные данные – это показатели классификации "Correctly Classified Instances" (59.1%) и "Incorrectly Classified Instances" (40.9%). Кроме того, следует обратить внимание на число в первой строке столбца ROC Area (0.616). Чуть позже мы подробно обсудим эти значения, пока же просто запомните их. Наконец, таблица Confusion Matrix показывает количество ложноположительных (516) и ложноотрицательных (710) распознаваний. Поскольку показатель точности нашей модели – 59,1%, то в первоначальном рассмотрении ее нельзя назвать достаточно хорошей.

Вы сможете увидеть дерево, если щелкнете правой кнопкой мышки в панели результирующей модели. В контекстном меню выберите опцию Visualize tree. На экране отобразится визуальное представление классификационного дерева нашей модели (рисунок 3), однако в данном случае картинка мало чем нам поможет. Еще один способ увидеть дерево модели – прокрутить вверх вывод в окне Classifier Output, там вы найдете текстовое описание дерева с узлами и листьями.

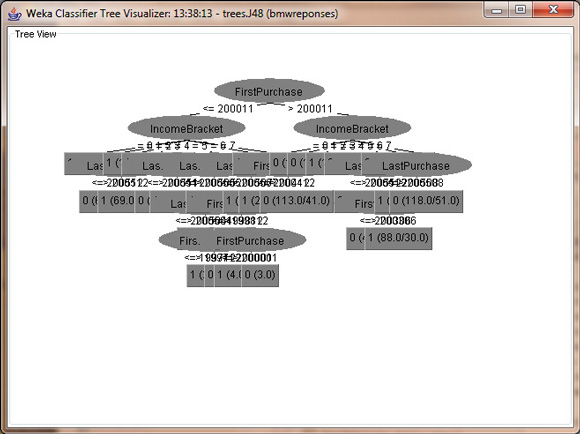


Рисунок 6 -Визуальное представление дерева классификации

Остался последний этап проверки классификационного дерева: нам надо пропустить оставшийся набор данных через полученную модель и проверить, насколько результаты классификации будут отличаться от реальных данных. Для этого в секции Test options выберите опцию Supplied test set и нажмите на кнопку Set. Укажите файл bmw-test.arff, содержащий оставшиеся 1500 данных, которые не были включены в обучающий набор. При нажатии на кнопку Start WEKA пропустит тестовые данные через модель и покажет результат работы модели. Давайте нажмем на Start и проверим, что у нас получилось.

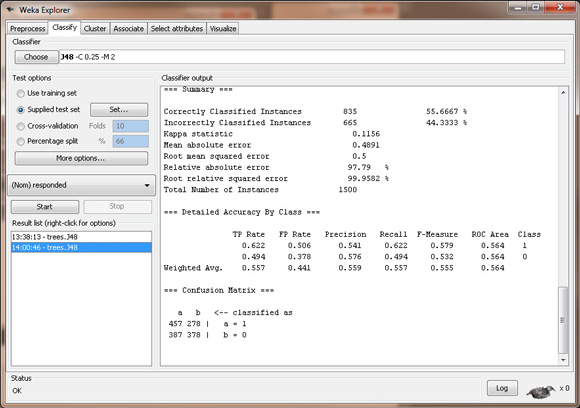


Рисунок 7 - Проверка классификационного дерева

Сравнивая показатель Correctly Classified Instances для тестового набора (55,7%) с этим же показателем для обучающего набора (59,1%), мы видим, что точность модели для двух разных наборов данных примерно одинакова. Это значит, что новые данные, которые будут использоваться в этой модели в будущем, не снизят точность ее работы.

Однако, поскольку собственно точность модели довольно низка (всего лишь 60% данных классифицировано верно), мы имеем полное право остановиться и сказать: «Она работает с точностью чуть выше 50%, с таким же успехом мы можем просто пытаться угадать значение случайным образом». Существуют случаи, когда использование алгоритмов интеллектуального анализа данных приводит к созданию неудачной аналитической модели.

Классификационная модель не подходит для анализа имеющихся у нас данных.

#### Метод ближайших соседей

Алгоритм метода ближайших соседей во многом схож с алгоритмом, используемым в методе кластеризации. Метод определяет расстояние между неизвестной точкой и всеми известными точками данных. Самый простой и наиболее распространенный способ определения расстояния – это нормализованное эвклидово расстояние.

Загрузим файл bmw-training.arff в WEKA, выполнив в закладке Preprocess.

Точно так же, как мы проделали это для методов регрессионного анализа и классификации в предыдущих статьях, мы должны открыть закладку Classify. В панели Classify нужно выбрать опцию lazy, а затем Ibk (здесь IB означает Instance-Based – обучение на примерах, а k указывает на количество соседей, поведение которых мы хотим исследовать). Убедитесь, что вы выбрали опцию Use training set, чтобы использовать набор данных, который мы только что загрузили в WEKA. Нажмите кнопку Start.

Результат вычислений IBk

=== Evaluation on training set ===

=== Summary ===

Correctly Classified Instances 2663 88.7667 %

Incorrectly Classified Instances 337 11.2333 %

Kappa statistic 0.7748

Mean absolute error 0.1326

Root mean squared error 0.2573

Relative absolute error 26.522 %

Root relative squared error 51.462 %

Total Number of Instances 3000

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure ROC Area Class

0.95 0.177 0.847 0.95 0.896 0.972 1

0.823 0.05 0.941 0.823 0.878 0.972 0

Weighted Avg. 0.888 0.114 0.893 0.888 0.887 0.972

=== Confusion Matrix ===

a b <-- classified as

1449 76 | a = 1

261 1214 | b = 0

У модели, использующей метод ближайших соседей, показатель точности равен 89% - совсем неплохо для начала, учитывая то, что точность предыдущей модели составляла всего 59%. Практически 90% точности – это вполне приемлемый уровень. Давайте рассмотрим результаты работы метода в терминах ложных определений, чтобы вы смогли на конкретном примере увидеть, как именно WEKA может использоваться для решения реальных вопросов бизнеса.

Результаты использования модели на нашем наборе данных показывают, что у нас есть 76 ложноположительных распознаваний (2.5%) и 261 ложноотрицательных распознаваний (8.7%). В нашем случае ложноположительное распознавание означает, что модель считает, что данный покупатель приобретет расширенную гарантию, хотя на самом деле он отказался от покупки. Ложноотрицательное распознавание, в свою очередь, означает, что согласно результатам анализа данный покупатель откажется от расширенной гарантии, а на самом деле он ее купил. Предположим, что стоимость каждой рекламной листовки, рассылаемой дилером, составляет $3, а покупка одной расширенной гарантии приносит ему 400$ дохода. Таким образом, ошибки ложного распознавание в терминах расходов и доходов нашего дилера будут выглядеть следующим образом: 400$ - (2.5% \* $3) - (8.7% \* 400) = $365. Следовательно, ложное распознавание ошибается в пользу дилера. Сравним этот показатель с данными модели классификации: $400 - (17.2% \* $3) - (23.7% \* $400) = $304. Как вы видите, использование более точной модели повышает потенциальный доход дилера на 20%.

#### Naive Bayes (наивный байесовский метод)

"Наивная" классификация - достаточно прозрачный и понятный метод классификации. "Наивной" она называется потому, что исходит из предположения о взаимной независимости признаков.

Свойства наивной классификации:

1. Использование всех переменных и определение всех зависимостей между ними.

2. Наличие двух предположений относительно переменных:

* все переменные являются одинаково важными;
* все переменные являются статистически независимыми, т.е. значение одной переменной ничего не говорит о значении другой.

Для использования этого метода в системе Weka панели Classify нужно выбрать опцию NativeBayes.

Результаты интерпретируются также как и при использовании других методов классификации.

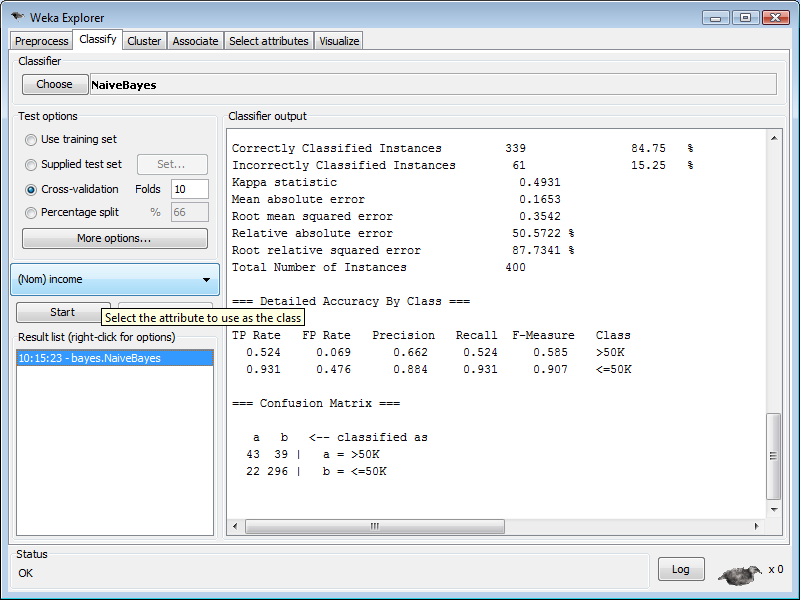


Рисунок 8 – Окно выбора метода

#### 1R

Метод классификации 1R – один из самых простых и понятных методов классификации. Применяется как к числовым данным, которые разбиваются на промежутки, так и к данным типа nominal.

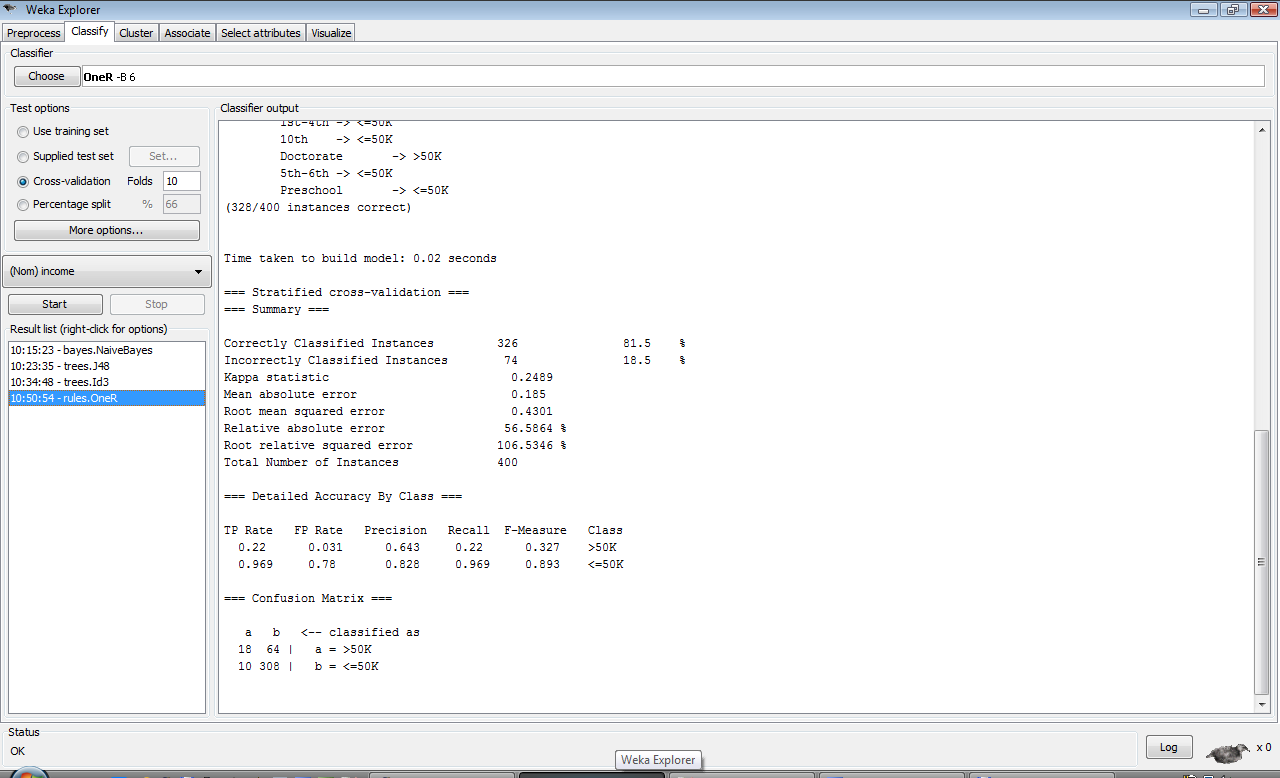


Рисунок 10

Для использования этого метода в системе Weka панели Classify нужно выбрать опцию OneR.

Результаты интерпретируются также как и при использовании других методов классификации.

#### SVM

Для этого метода не требуется каких-либо преобразований исходной выборки.

Данный метод является алгоритмом классификации с использованием математических функций. Метод использует нелинейные математические функции. Номинальные данные преобразуются в числовые. Основная идея метода опорных векторов – перевод исходных векторов в пространство более высокой размерности и поиск максимальной разделяющей гиперплоскости в этом пространстве.

Для использования этого метода в системе Weka панели Classify нужно выбрать опцию SMO.

Результаты интерпретируются также как и при использовании других методов классификации.

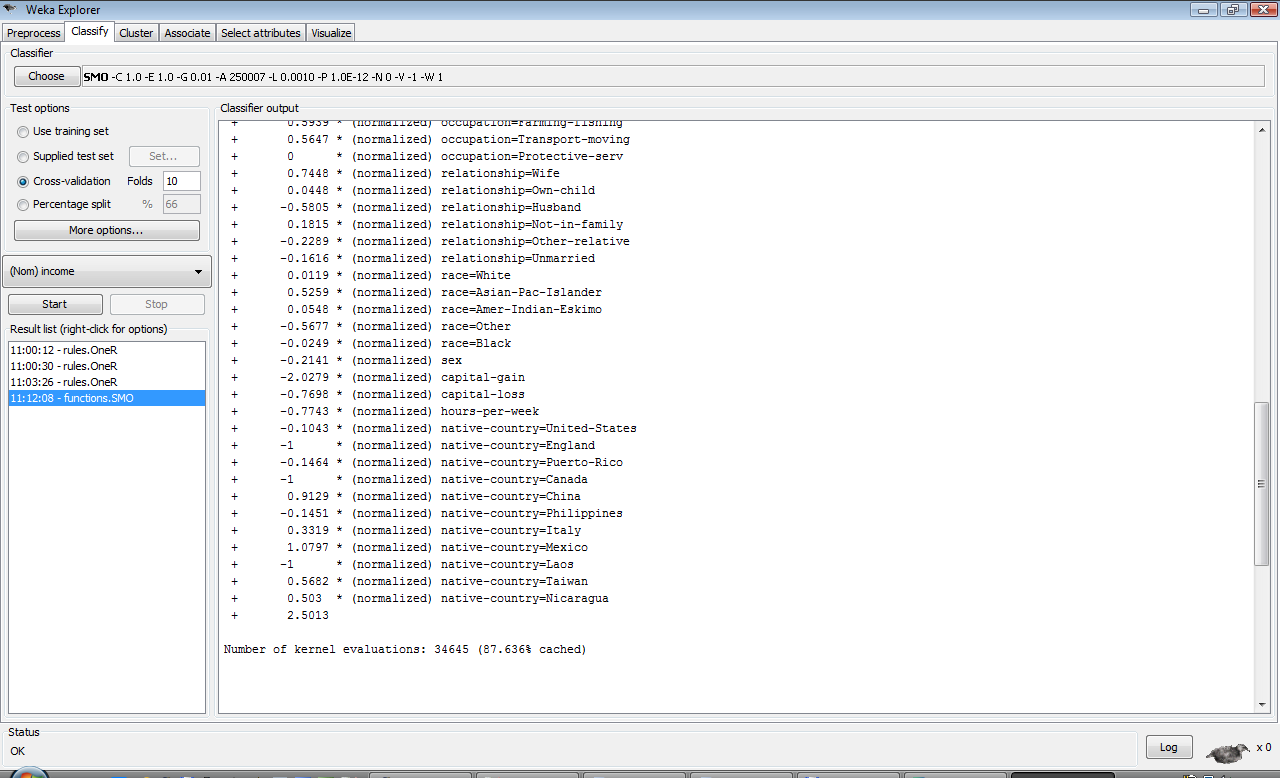


Рисунок 11

### Кластеризация

Кластеризация позволяет разбить данные на группы, каждая из которых имеет определенные признаки. Метод кластерного анализа используется в тех случаях, когда необходимо выделить некоторые правила, взаимосвязи или тенденции в больших наборах данных. В зависимости от потребностей бизнеса, вы можете выделить несколько различных групп данных. Одно из явных преимуществ кластеризации по сравнению с классификацией состоит в том, что для разбиения множества на группы может использоваться любой атрибут (если вы помните, метод классификации использует только определенное подмножество атрибутов). В качестве основного недостатка метода кластеризации следует упомянуть тот факт, что составитель модели должен заранее решить, на сколько групп следует разбить данные. Для человека, который не имеет никакого представления о конкретном наборе данных, принять такое решение достаточно затруднительно. Следует ли нам создать три группы или пять групп? А может, нам нужно определить десять групп? Может потребоваться несколько итераций проб и ошибок, для того чтобы определить оптимальное количество кластеров.

Тем не менее, для среднестатистического пользователя кластеризация может оказаться наиболее полезным методом интеллектуального анализа данных. Этот метод позволит вам быстро разбить ваши данные на отдельные группы и сделать конкретные выводы и предположения относительно каждой группы. Математические методы, реализующие кластерный анализ, довольно сложны и запутаны, так что в случае кластеризации мы будем целиком полагаться на вычислительные возможности WEKA.

Загрузите файл bmw-browsers.arff в WEKA, выполнив те же шаги, которые мы проделали ранее для открытия данных в закладке **Preprocess**.

Поскольку мы хотим разбить имеющиеся у нас данные на кластеры, вместо закладки **Classify** нам потребуется закладка Cluster. Нажмите на кнопку **Choose** и в предлагаемом меню выберите опцию SimpleKMeans.

Щелкните на опции SimpleKMeans (дизайн пользовательского интерфейса оставляет желать лучшего, но работать с ним можно). Единственный атрибут алгоритма, который нас интересует – это поле numClusters, которое указывает на количество кластеров для разбиения (напоминаем, что это значение вам нужно выбрать еще до создания модели). Изменим значение по умолчанию (2) на 5.

Результаты кластеризации

Cluster#

Attribute Full Data 0 1 2 3 4

(100) (26) (27) (5) (14) (28)

==================================================================================

Dealership 0.6 0.9615 0.6667 1 0.8571 0

Showroom 0.72 0.6923 0.6667 0 0.5714 1

ComputerSearch 0.43 0.6538 0 1 0.8571 0.3214

M5 0.53 0.4615 0.963 1 0.7143 0

3Series 0.55 0.3846 0.4444 0.8 0.0714 1

Z4 0.45 0.5385 0 0.8 0.5714 0.6786

Financing 0.61 0.4615 0.6296 0.8 1 0.5

Purchase 0.39 0 0.5185 0.4 1 0.3214

Clustered Instances

0 26 ( 26%)

1 27 ( 27%)

2 5 ( 5%)

3 14 ( 14%)

4 28 ( 28%)

Данные кластеризации показывают, каким образом сформирован каждый кластер: значение «1» означает, что у всех данных в этом кластере соответствующий атрибут равен 1, а значение «0» означает, что у всех данных в этом кластере соответствующий атрибут равен 0. Данные соответствуют среднему значению атрибута на кластере. Каждый кластер характеризует определенный тип поведения клиентов, таким образом, на основании нашего разбиения мы можем сделать некоторые полезные выводы:

* Кластер 0— эту группу посетителей можно было бы назвать «мечтатели». Они бродят вокруг дилерского центра, рассматривая машины, выставленные на внешней парковке, но никогда не заходят внутрь, и, хуже того, никогда ничего не покупают.
* Кластер 1— эту группу следовало бы назвать «поклонники М5», поскольку они сразу же подходят к выставленным автомобилям этой модели, полностью игнорируя BMW серии 3 или Z4. Тем не менее, эта группа не отличается высокими показателями покупки машин – всего 52%. Это потенциально может свидетельствовать о недостаточно продуманной стратегии продаж и о необходимости улучшить работу дилерского центра, например, за счет увеличения количества продавцов в секции M5.
* Кластер 2— эта группа настолько мала, что мы могли бы назвать ее выбраковкой. Дело в том, что данные этой группы статистически довольно разбросаны, и мы не можем сделать каких-либо определенных заключений относительно поведения посетителей, попавших в этот кластер (подобная ситуация может указывать на то, что вам следует сократить количество кластеров в модели)
* Кластер 3— эту группу следовало бы назвать «любимцы BMW», потому что посетители, попавшие в это кластер, всегда покупают машину и получают необходимое финансирование. Обратите внимание, данные этого кластера демонстрируют интересную модель поведения этих покупателей: сначала они осматривают выставленные на парковке машины, а затем обращаются к поисковой системе дилерского центра. Как правило, они покупают модели M5 или Z4, но никогда не берут модели третьей серии. Данные этого кластера указывают на то, что дилерскому центру следует активнее привлекать внимание к поисковым компьютерам (может быть, вынести их на внешнюю парковку), и кроме того, следует найти какой-нибудь способ выделить модели M5 и Z4 в результатах поиска, чтобы гарантированно обратить на них внимание посетителей. После того, как посетитель, попавший в этот кластер, выбрал определенную модель автомобиля, он гарантированно получает необходимый кредит и совершает покупку.
* Кластер 4— эту группу можно назвать «начинающие владельцы BMW», поскольку они всегда ищут модели 3 серии и никогда не интересуются более дорогими M5. Они сразу же проходят в демонстрационный зал, не тратя время на осмотр машин на внешней стоянке. Кроме того, они не пользуются поисковой системой центра. Примерно 50% этой группы получают одобрение по кредиту, тем не менее, покупку совершают всего 32% участников. Анализируя данные этого кластера, можно сделать следующий вывод: посетители этой группы хотели бы купить свой первый BMW и точно знают, какая машина им нужна (модель 3 серии с минимальной конфигурацией). Однако, для того чтобы купить машину, им нужно получить положительное решение по кредиту. Чтобы повысить уровень продаж среди посетителей 4 кластера, дилерскому центру следовало бы понизить уровень требований для получения кредита или снизить цены на модели 3 серии.

Еще один интересный способ изучения результатов кластеризации – это визуальное представление данных. Щелкните правой кнопкой мышки в секции Result List закладки Cluster (и вновь элементы пользовательского интерфейса оставляют желать лучшего). В контекстном меню выберите опцию Visualize Cluster Assignments. В результате откроется окно с графическим представлением результатов кластеризации, настройки которого вы можете выбрать наиболее удобным для вас образом. Для нашего примера, измените настройку оси X так, чтобы она соответствовала количеству автомобилей М5 (M5 (Num)), а настройку оси Y – так, чтобы она показывала количество купленных автомобилей (Purchase (Num)), и укажите выделение каждого кластера отдельным цветом (для этого установите значение поля Color в Cluster (Nom)). Такие настройки помогут нам оценить распределение по кластерам в зависимости от того, сколько человек интересовалось BMW M5, и сколько человек купило эту модель. Кроме того, сдвиньте указатель Jitter примерно на три четверти в сторону максимума, это искусственным образом увеличит разброс между группами точек, чтобы вам было удобнее их просматривать.

Соответствует ли визуальное отображение кластеризации тем заключениям, которые мы сделали на основании данных в листинге 5? Как мы видим, в окрестности точки X=1, Y=1 (посетители, которые интересовались автомобилями модели M5 и купили их) расположены только два кластера: 1 и 3. Аналогично, в окрестности точки X=0, Y=0 расположены только два кластера: 4 и 0. Соответствует ли это нашим выводам? Да, соответствует. Кластеры 1 и 3 покупают BMW M5, в то время как кластер 0 не покупает ничего, а кластер 4 ищет BMW серии 3. На рисунке 8 показано визуальное отображение кластеров нашей модели. Мы предлагаем вам самостоятельно попрактиковаться в обнаружении других трендов и течений, меняя настройки осей X и Y.

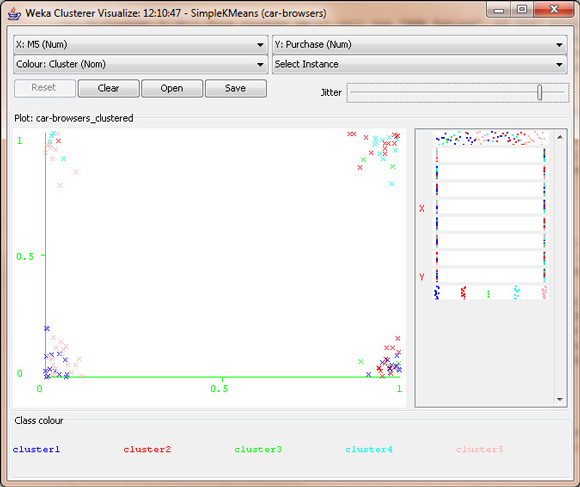


Рисунок 12 - Визуальное отображение кластеризации

### Построение ассоциативных правил

Нахождение ассоциативных правил происходит почти так же, как и классификация. На вкладке Associate выбирается метод нахождения, для него выставляются параметры кликом на его названии, после чего нажимается кнопка Start и анализируется вывод (перед началом использования метода Априори необходимо применить фильтр RemoveType и удалить numeric-атрибуты). В нашем случае ассоциативные правила строятся по методу Априори.

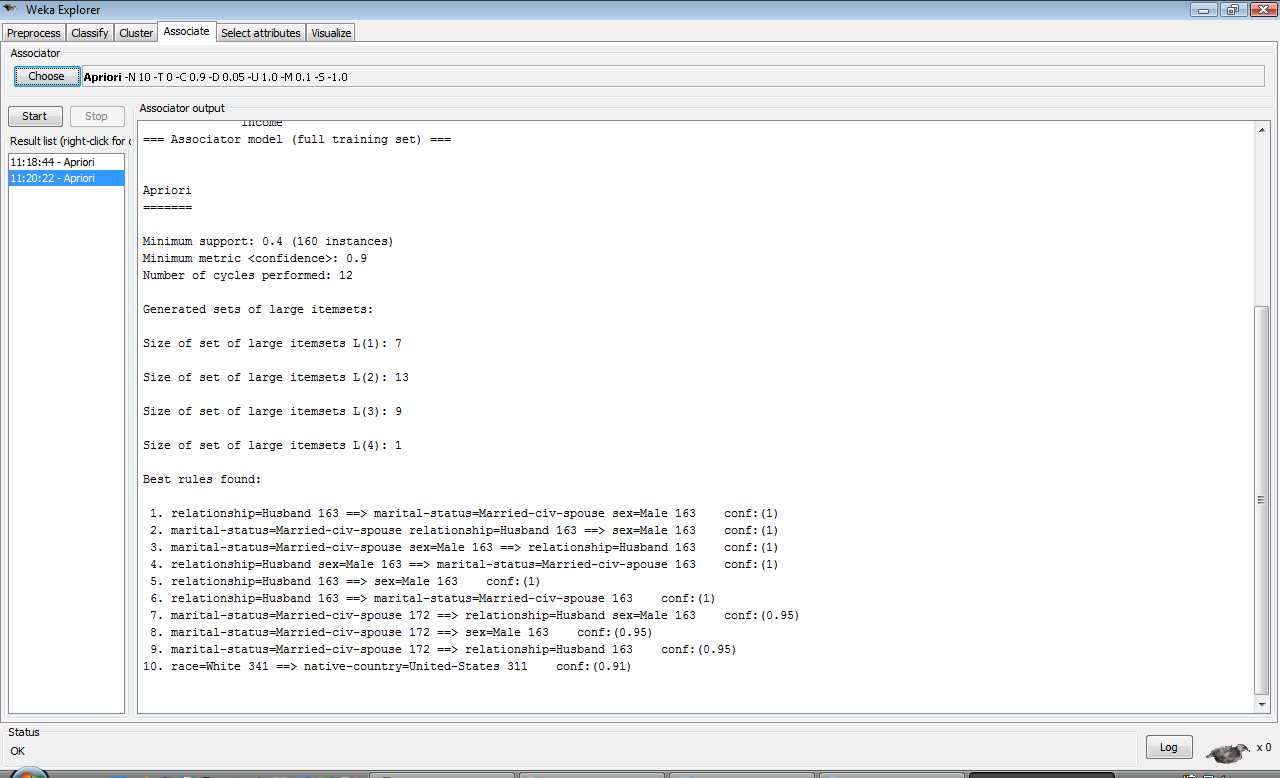


Рисунок 13

# Варианты задания

Наборы данных можно брать из открытых источников (наборы не должны повторяться) или генерировать самостоятельно по вариантам:

1. Погодные условия в регионе
2. Продажа комплектующих изделий
3. Демографическая ситуация в регионе
4. Продажа земельных участков
5. Рынок труда
6. Больница
7. Железнодорожный транспорт
8. Авиационные перевозки
9. Олимпиада
10. Футбол
11. Туристический бизнес
12. Социальные сети
13. Интернет-провайдер
14. Здравоохранение
15. Автострахование
16. Кредитование
17. Экология
18. Правонарушения
19. Литература
20. Компьютеры