МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М.В. ЛОМОНОСОВА

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

КАФЕДРА МАТЕМАТИЧЕСКИХ МЕТОДОВ ПРОГНОЗИРОВАНИЯ

Задание №5(Работа с пакетом RWeka системы R)

Отчет о проделанной работе

Евгений Зак

317 группа

Москва 2012

Что мне нравится в стандартах, так это то,
что их всегда очень много, и я могу
выбирать из огромного списка.
Andrew S. Tannenbaum

1.Постановка задачи.

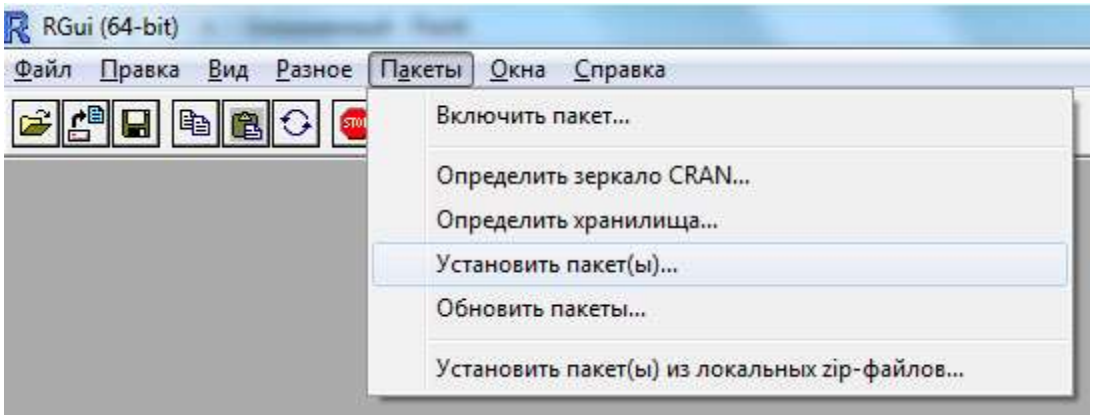
Автору задания было предложено разобраться с одним из пакетов системы R и написать некий отчет-мануал по этому пакету.

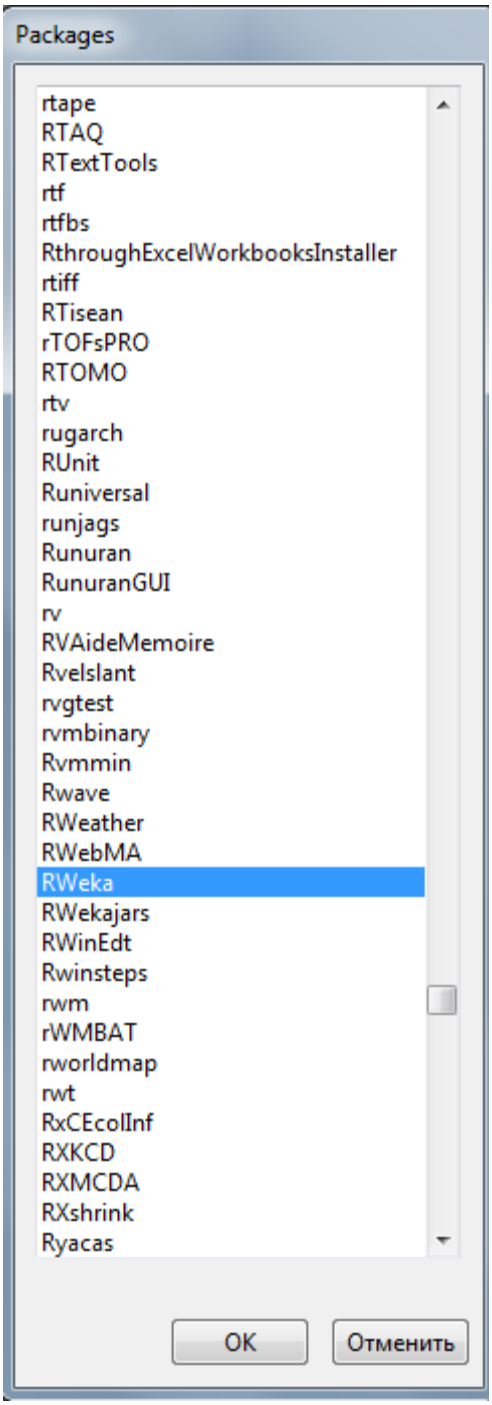
Пакет Rweka был выбран не случайно. Пакет представляет собой интерфейс для доступа из R к алгоритмам, которые заложены в WEKA. А этот пакет обладает большим количеством алгоритмов для реализации многих задача распознавания образов (классификация, восстановление регрессии, кластеризация и т.д.).

Забегая вперед можно сказать, что пакет сделан качественно и понятно для пользователя. Он содержит множество методов. В данном отчете представлены лишь основные, т.к. описать все функции очень сложно не имея богатого опыта использования каждой из них на реальных задачах.

2.Начало работы с пакетом RWeka.

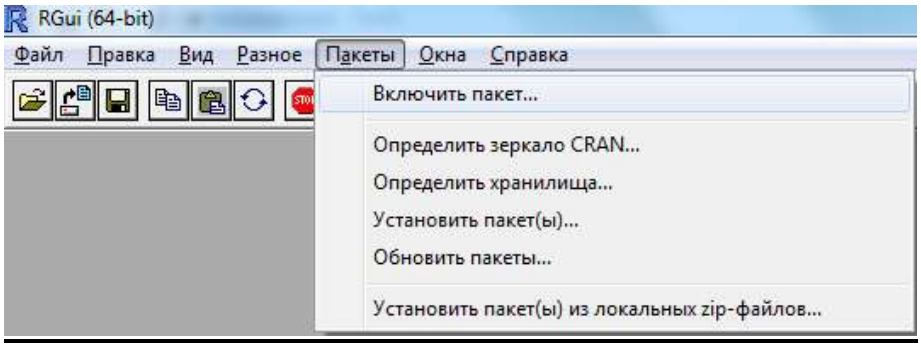
Для того, чтобы начать работать с пакетом, необходимо его загрузить. Сделать это можно как отдельно, так и в самом R:





Для полноценной работы пакета, также необходимо загрузить пакеты rJava и RWekajars.

Потом установленные пакеты необходимо подключить. Сделать это можно так:



Таким образом можно подключать любые пакеты. В данном отчете также используется пакет «**party**». Он нужен для визуализации решающих деревьев. Так же при использовании пакета может понадобиться пакет «**Rgraphviz**».

3.Функции пакета RWeka.

Прежде чем начать что-либо классифицировать, необходимо загрузить данные. Система Weka работает с данными формата .arff (Attribute-Relation File Format), и считать такие файлы можно с помощью функции **read.arff**

Синтаксис

```
> read.arff(file)
```

Аргументы

file – строковая переменная, содержащая полный путь к файлу.

Пример

```
> read.arff("E:\\zak.arff")
```

Сразу скажем, что данные можно сохранить(например после нормировки или других изменений). Для этого в пакете предусмотрена функция **write.arff**

Синтаксис

```
> write.arff(x, file, eol = "\n")
```

Аргументы

x – переменная, содержащая данные для сохранения;

file – строковая переменная, содержащая полный путь к файлу, или “” для вывода в стандартный поток вывода;

eol – символ(ы) для записи в конце каждой строки файла (по умолчанию \n).

Пример

```
> write.arff(zak , "E:\\zak.arff", "\n" )
```

Возникает вопрос, а как можно изменить данные? Weka обладает большим набором фильтров для обработки данных. Например, **Normalize** или **Discretize**

Синтаксис

> Normalize(formula, data, ...)

> Discretize(formula, data, …)

Аргументы

formula – название признака(ов), к которым необходимо применить фильтр(или ~. для применения ко всем признакам);

data – переменная, которая содержит данные, полученные с помощью команды read.arff. можно использовать стандартные данные, которые идут вместе с пакетом RWeka;

далее в параметрах функций можно указывать необходимые параметры. У каждого фильтра свой набор, есть такие, как na.action (реакция на пропуски данных в файле) и т.д.

Пример

> w <- read.arff(system.file("arff","weather.arff", package = "RWeka"))

> Normalize(~., data = w)

> Discretize(play ~., data = w)

Теперь, когда читатель умеет считывать, обрабатывать и записывать данные уместно описать функции для классификации. Посмотреть весь список установленных пакетов можно с помощью вызова команды **WPM("list-packages", "installed").** После ее вызова пользователь получает информацию о всех пакетах, установленных на компьютере, их версиях и описании.

Пример

>WPM("list-packages", "installed")

Installed Repository Package
========= ========== =======
----- 1.0.1 J48graft: Class for generating a grafted (pruned or unpruned) C4.5 decision tree

----- 1.0.0 JDBCDriversDummyPackage: Dummy package that provides a place to drop JDBC driver jar files so that they get loaded by the system.
----- 1.0.1 LVQ: Cluster data using the Learning Vector Quantization algorithm.
----- 1.0.1 LibLINEAR: A wrapper class for the liblinear tools
----- 1.0.3 LibSVM: A wrapper class for the libsvm tools
----- 1.0.1 NNge: Nearest-neighbor-like algorithm using non-nested generalized exemplars (which are hyperrectangles that can be viewed as if-then rules)
----- 1.0.0 PCP: Parallel Coordinates Plot
----- 1.0.2 RBFNetwork: Classes that implements radial basis function networks

Список пакетов можно изменять: удалять или загружать новые
>WPM("remove-packages", "repository", "XMeans")

> WPM("load-packages", "repository", "randomForest")

Выбрав из списка тот метод, который нужен пользователю его можно использовать с помощью функции, которая имеет имя названием метода, например так: **LinearRegression**

Синтаксис

> LinearRegression(formula, data, …)

Аргументы

formula – название целевого признака;

data – переменная, которая содержит данные, полученные с помощью команды read.arff. можно использовать стандартные данные, которые идут вместе с пакетом RWeka;

далее в параметрах функций можно указывать необходимые параметры. У каждой функции свой набор, есть такие как na.action (реакция на пропуски данных в файле) и т.д.

Пример

> LinearRegression(weight ~ feed, data = chickwts)

> AdaBoostM1(Species ~ ., data = iris)

> DF1 <- read.arff(system.file("arff", "contact-lenses.arff",

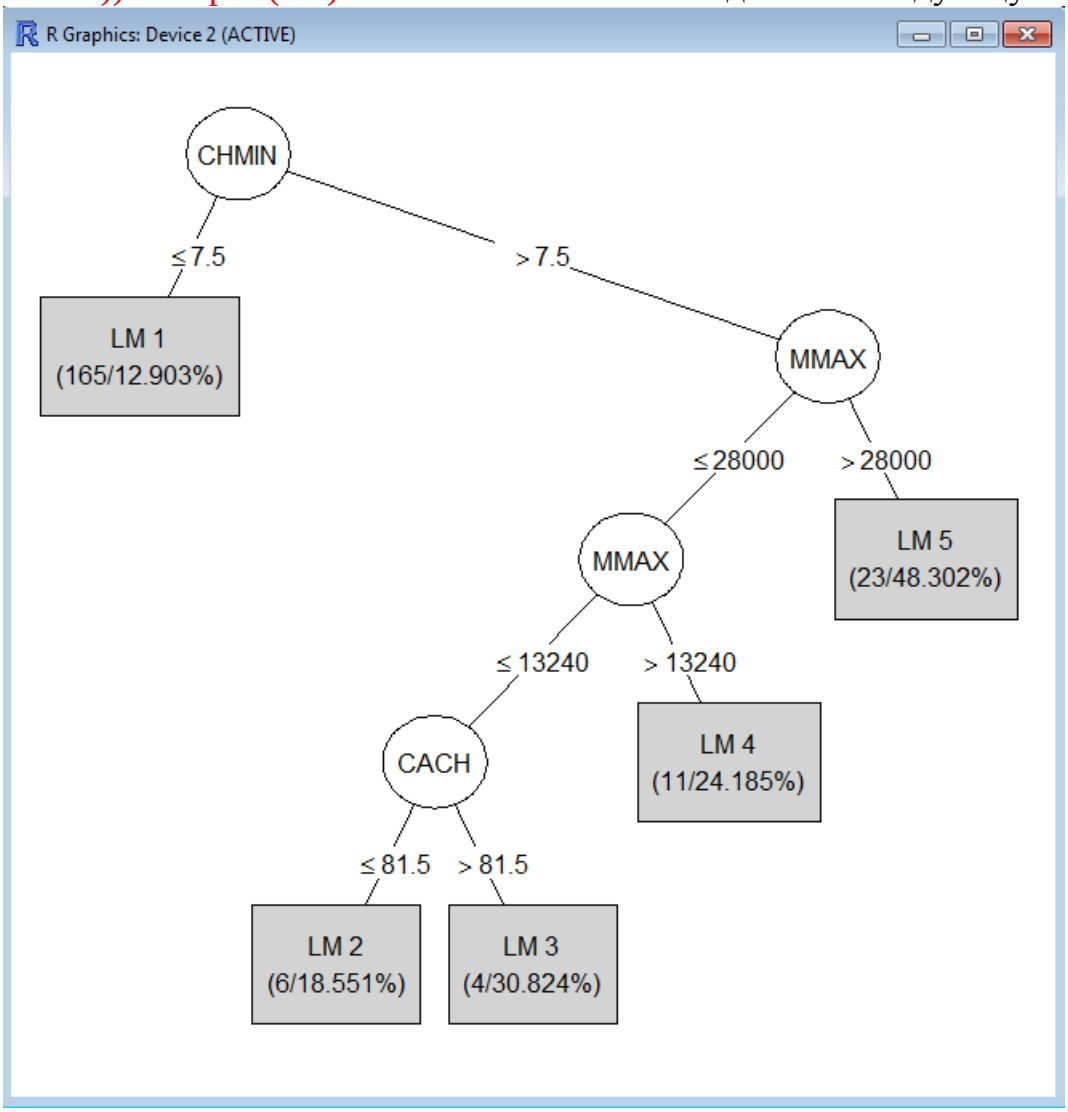
```
package = "RWeka"))

> LinearRegression (‘contact-lenses‘ ~ ., data = DF1)

> DF2 <- read.arff(system.file("arff", "contact-lenses.arff",
package = "RWeka"))

> J48(‘contact-lenses‘ ~ ., data = DF2)

> DF3 <- read.arff(system.file("arff", "cpu.arff", package = "RWeka"))

> m <- M5P(class ~ ., data = DF3)
```

Имеет смысл рассмотреть последнюю строку примера. В переменную m запишется результат и данные о модели классификации. Если попросить вывести m, то пользователь может увидеть следующее:

```
M5 pruned model tree:

(using smoothed linear models)

CHMIN <= 7.5 : LM1 (165/12.903%)

CHMIN > 7.5 :

| MMAX <= 28000 :

| | MMAX <= 13240 :

| | | CACH <= 81.5 : LM2 (6/18.551%)

| | | CACH > 81.5 : LM3 (4/30.824%)

| | MMAX > 13240 : LM4 (11/24.185%)

| MMAX > 28000 : LM5 (23/48.302%)

LM num: 1

class =

-0.0055 * MYCT

+ 0.0013 * MMIN

+ 0.0029 * MMAX

+ 0.8007 * CACH

+ 0.4015 * CHMAX

+ 11.0971

LM num: 2

class =

-1.0307 * MYCT

+ 0.0086 * MMIN

+ 0.0031 * MMAX

+ 0.7866 * CACH

- 2.4503 * CHMIN

+ 1.1597 * CHMAX

+ 70.8672

LM num: 3

class =

-1.1057 * MYCT

+ 0.0086 * MMIN

+ 0.0031 * MMAX

+ 0.7995 * CACH

- 2.4503 * CHMIN

+ 1.1597 * CHMAX

+ 83.0016

LM num: 4

class =

-0.8813 * MYCT

+ 0.0086 * MMIN
```

+ 0.0031 * MMAX	LM num: 5	- 1.3252 * CHMIN
+ 0.6547 * CACH	class =	+ 3.3671 * CHMAX
- 2.3561 * CHMIN	-0.4882 * MYCT	- 51.8474
+ 1.1597 * CHMAX	+ 0.0218 * MMIN	
+ 82.5725	+ 0.003 * MMAX	Number of Rules : 5
	+ 0.3865 * CACH	

Подключив пакет «**party**», можно визуализировать нашу модель. Например, если пользователь введет команду > if(require("party", quietly = TRUE)) plot(m3) то R выведет следующую картинку:



4.Литература.

1. http://alexanderdyakonov.narod.ru/upR.pdf
2. http://cran.gis-lab.info/web/packages/RWeka/RWeka.pdf
3. http://weka.wikispaces.com