Изготвили:

Сияна славова, 24963

Иван Капукаранов, 24958

Проект за курса „Откриване на знания в данни“

Задача: wine quality classification

# Декларация за липса на плагиатство

1. Тази курсова работа е моя работа, като всички изречения, илюстрации и програми от други хора са изрично цитирани.

2. Тази курсова работа или нейна версия не са представени в друг университет или друга учебна институция.

3. Разбирам, че ако се установи плагиатство в работата ми ще получа оценка “Слаб”.

Иван Капукаранов, ФН: 24958, 1 курс, ИИОЗ

Сияна Славова, ФН: 24963, 1 курс, ИИОЗ

Съдържание

[Декларация за липса на плагиатство 1](#_Toc455090839)

[Мотивация, Задача на курсовата работа 3](#_Toc455090840)

[Съществуващо решение 3](#_Toc455090841)

[Избрано решение 3](#_Toc455090842)

[Програмна реализация 3](#_Toc455090843)

[Предварителна обработка 3](#_Toc455090844)

[Анализ на данните 3](#_Toc455090845)

[Червено вино (Атрибут ~ Качество) 3](#_Toc455090846)

[Бяло вино (Атрибут ~ Качество) 4](#_Toc455090847)

[Реализация 5](#_Toc455090848)

[Резултати от експерименти 5](#_Toc455090849)

[Random forest резултати 6](#_Toc455090850)

[Обучаване с бяло вино(всички атрибути) тестване с бяло вино 6](#_Toc455090851)

[Обучаване с бяло вино(избрани атрибути) тестване с бяло вино 7](#_Toc455090852)

[Обучаване с червено вино(всички атрибути) тестване с червено вино 8](#_Toc455090853)

[Обучаване с червено вино(избрани атрибути) тестване с червено вино 9](#_Toc455090854)

[Смесено: обучаване с бяло вино(всички атрибути) тестване с червено вино 9](#_Toc455090855)

[Смесено: обучаване с червено вино(всички атрибути) тестване с бяло вино 10](#_Toc455090856)

[Смесено: обучаване с червено и бяло вино(всички атрибути) тестване с бяло вино 10](#_Toc455090857)

[Neural Networks резултати 11](#_Toc455090858)

[Обучаване с бяло вино(всички атрибути) тестване с бяло вино 11](#_Toc455090859)

[Обучаване с червено вино(всички атрибути) тестване с червено вино 11](#_Toc455090860)

[Смесено: обучаване с червено и бяло вино(всички атрибути) тестване с бяло вино 11](#_Toc455090861)

[Смесено: обучаване с червено и бяло вино(избрани атрибути) тестване с бяло вино 11](#_Toc455090862)

[Заключение и бъдещо развитие 11](#_Toc455090863)

[Разпределение на задачите 11](#_Toc455090864)

[Литература и използвани източници 12](#_Toc455090865)

# Мотивация, Задача на курсовата работа

Данните представлява две множества с информация за химическия състав на различни видове вино. Едното множество е за бяло вино, а другото - за червено. Един екземпляр се определя със следните атрибути:

fixed acidity

volatile acidity

citric acid

residual sugar

chlorides

free sulfur dioxide

total sulfur dioxide

density

pH

sulphates

alcohol

Целта е да се класифицира виното според неговото качество. Категориите за качество са общо 10, но реално и в двете множества данните са с качество между 3 и 9.

# Съществуващо решение

В намереното [съществуващо решение](http://www.r-bloggers.com/predicting-wine-quality-using-random-forests/) е използван алгоритъмът Random forest. Той е приложен само върху White Wine dataset.

Като предварителна обработка качеството на виното се разпределя в три категории:

* Лошо (с качеството, по – малко от 6)
* Нормално (с качество, което е равно на 6)
* Добро (с качество, по – голямо от 6)

Тази предварителна обработка е направена поради липсата на най – много обучаващи данни с качество 6 и доста по – малко в другите категории.

В това решение ефективността на алгоритъма е измерена като accuracy, която е приблизително 71,5%.

# Избрано решение

Ние избрахме да сравним представянето на два алгоритъма върху така поставената задача, а именно random forest и neural network. За реализация използвахме [R Project](https://www.r-project.org/). За разлика от предварителното разделяне в три категории на качеството в съществуващото решение, ние решихме да работим с оригиналните степени на качество от 1 до 10.

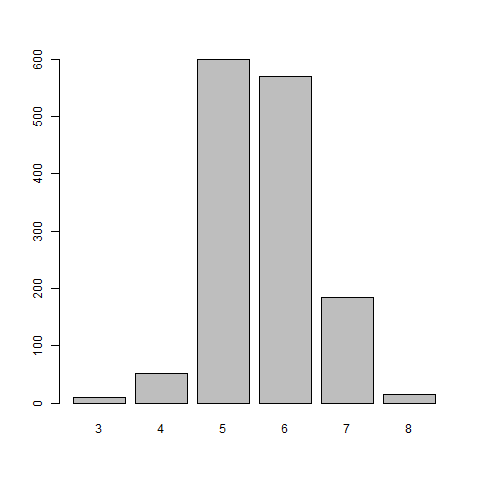
# Програмна реализация

## Предварителна обработка

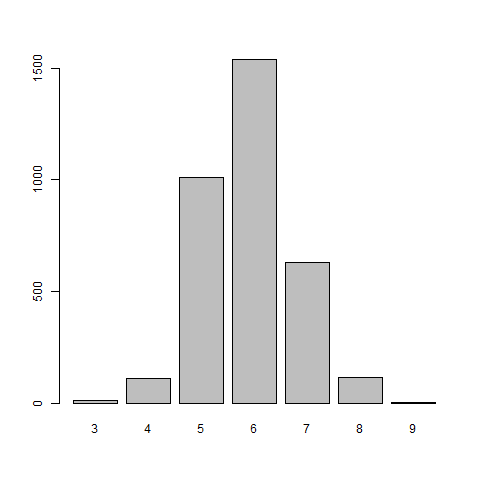
Тъй като множествата, които използвахме за задачата не бяха разделени на тестово и обучаващо, ние отделихме 10% от всяко едно от тях за тестване и 90% за обучение.

## Анализ на данните

Както се вижда от двете графики на разпределенията (Фигура 1 и Фигура 2), данните и за бяло и за червено вино са нормално разпределени със сравнително малко стандартно отклонение, което води на струпване на данни за средна степен на качество (степен 5 или степен 6). Вижда се също, че има много малко на брой обучаващи данни за екстремни степени на качество (например много ниско или много високо). От графиките се вижда и че реално степените на качество в множествата са от порядъка на 3 до 9.

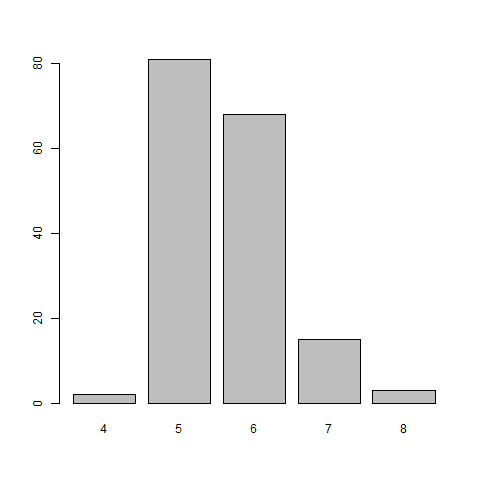


Фигура 1:Разпределение обучаващо множество – червено вино

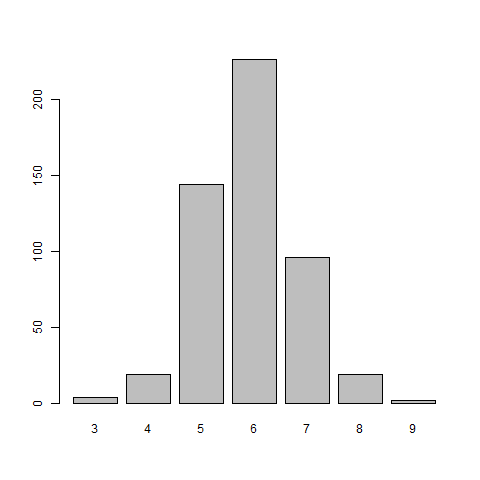


Фигура 2:Разпределение обучаващо множество– бяло вино

Ситуацията е подобна и при разпределението на тестовите множества (Фигура 3 и Фигура 4).



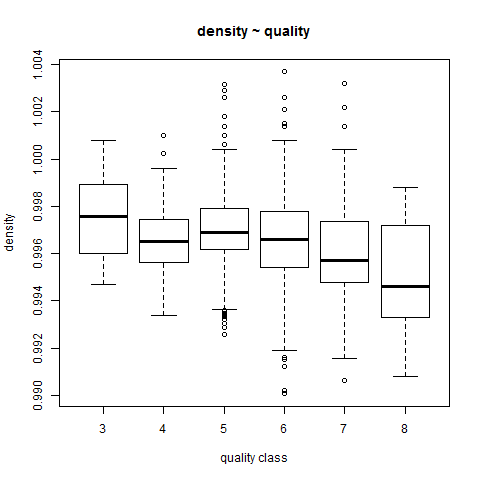
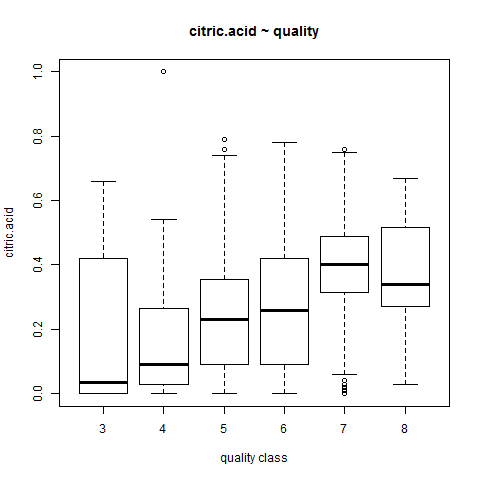
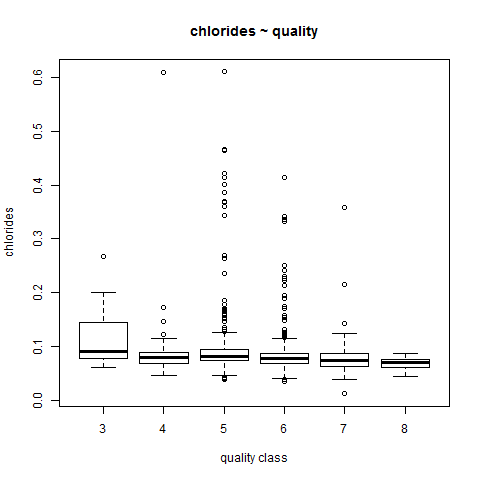
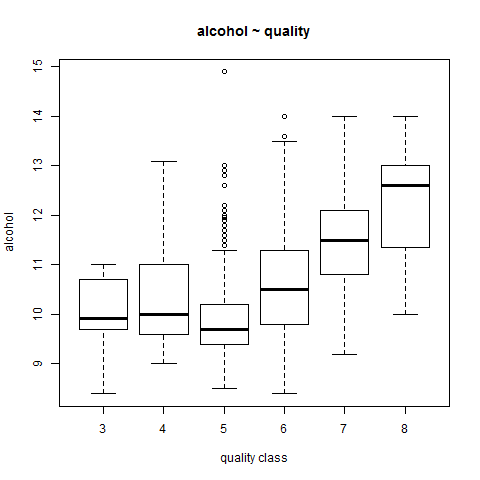
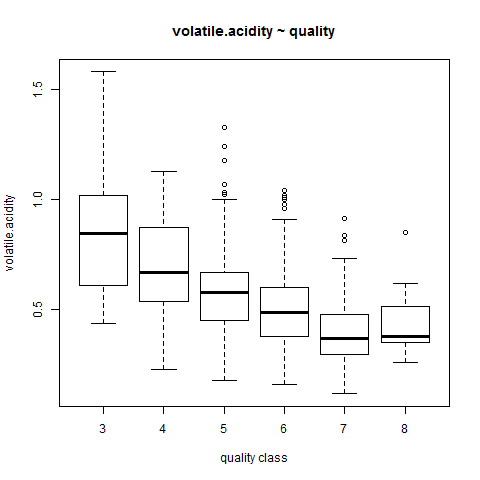
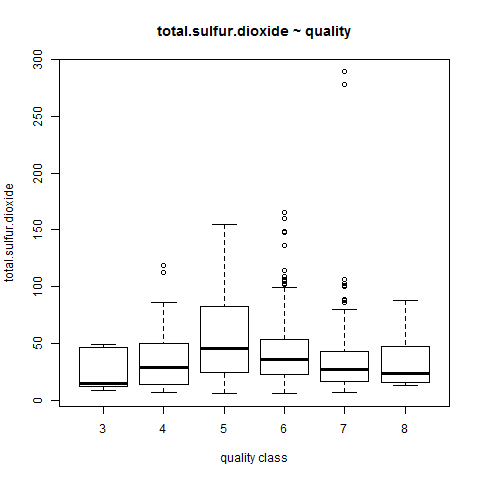
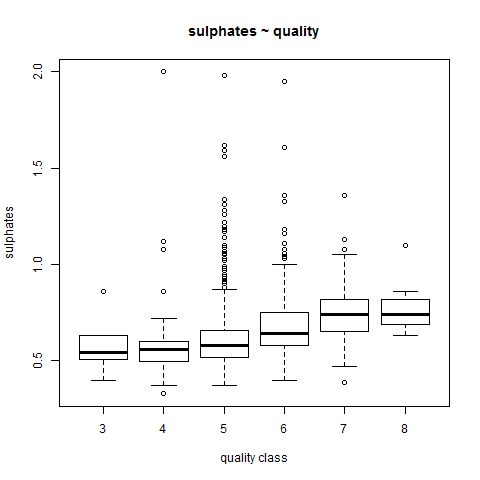
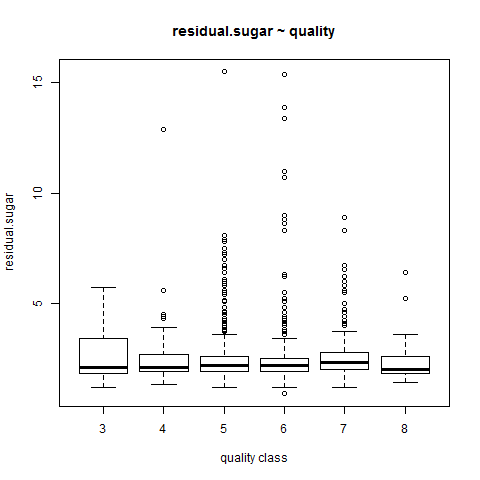
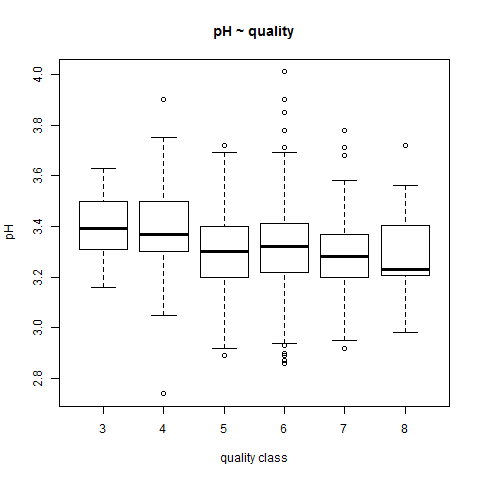
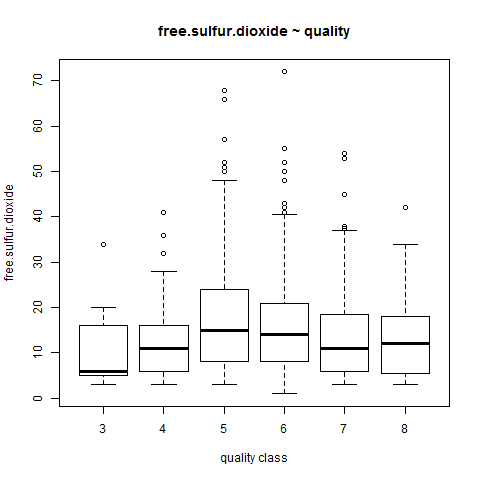
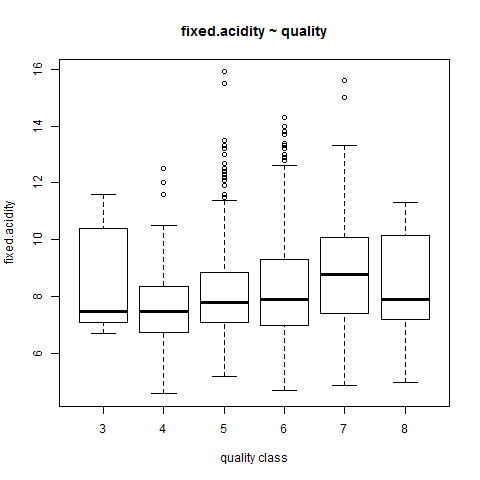
Фигура 3: Разпределение тестово множество - червено вино



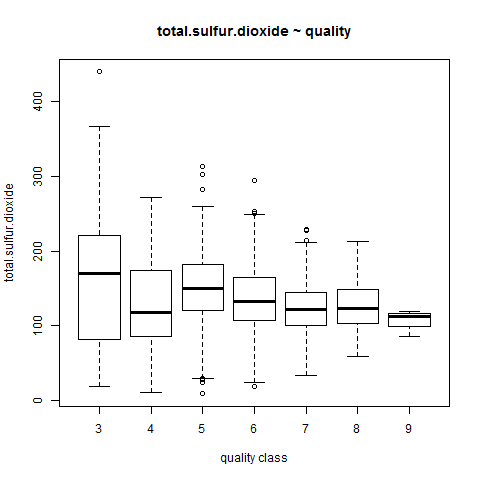
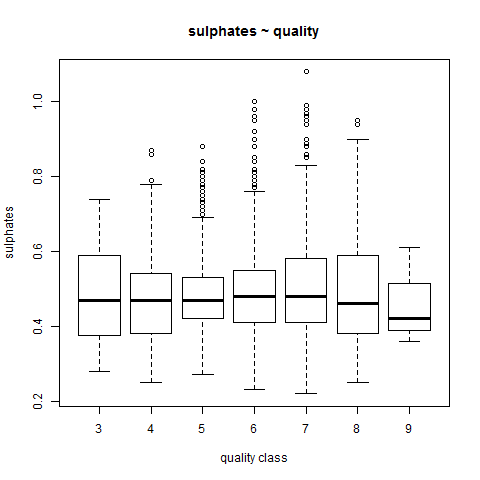
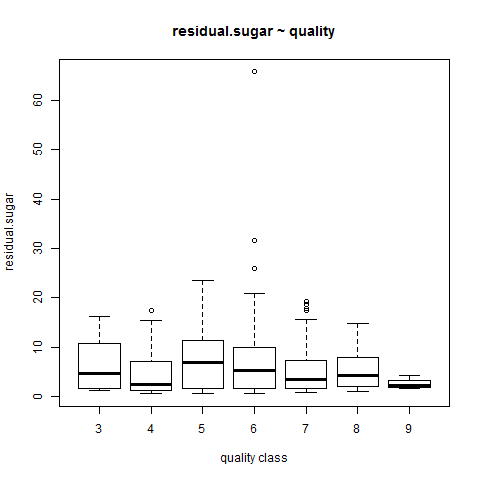
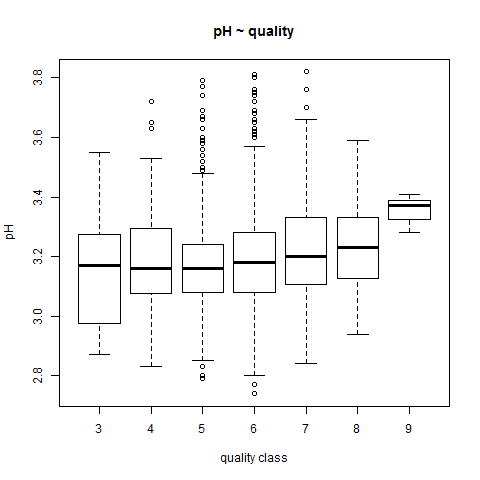
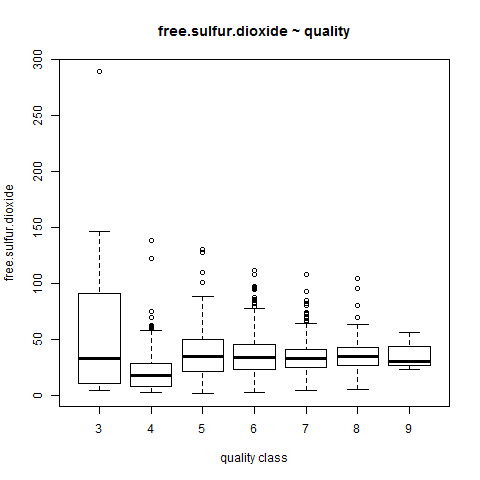
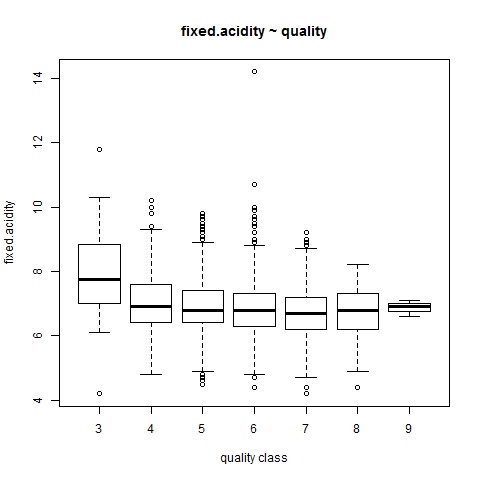
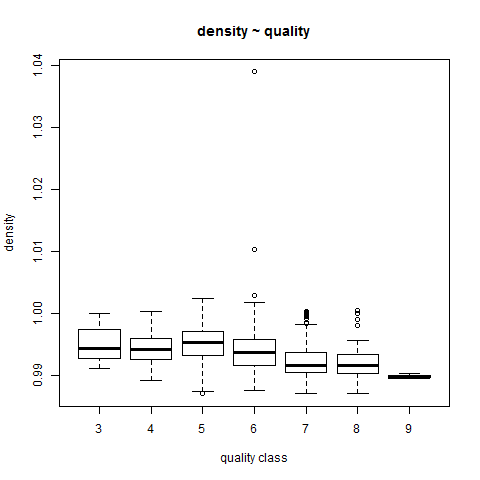
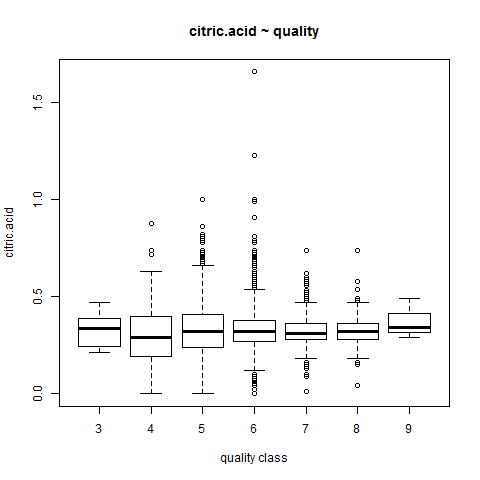
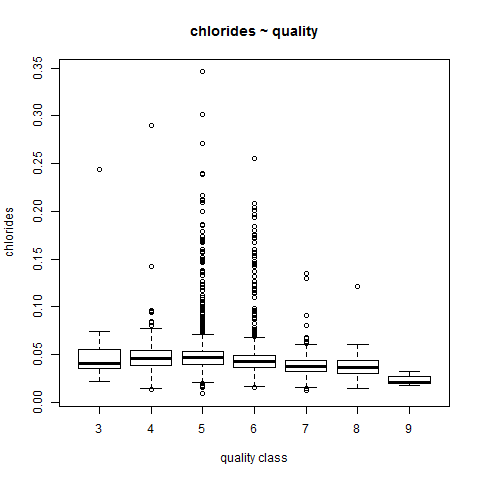
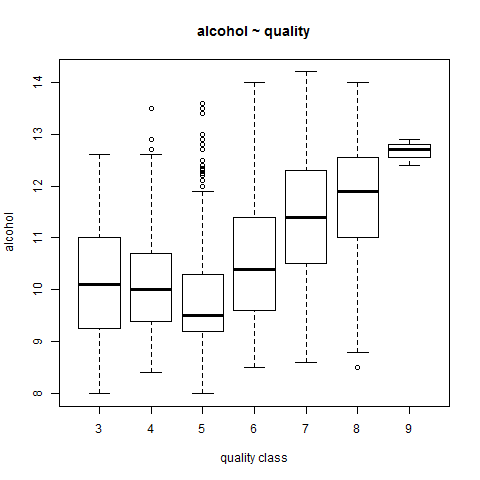
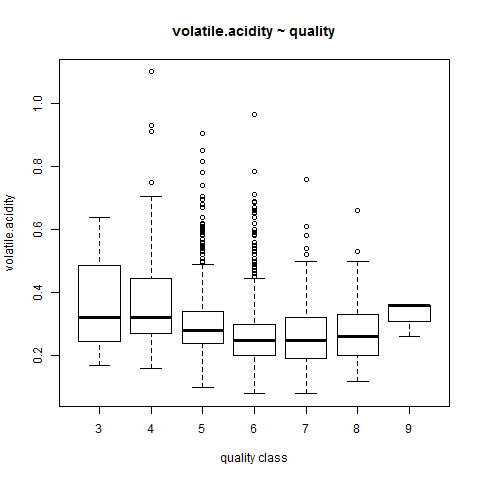
Фигура 4: Разпределение тестово множество - бяло вино

При анализа на данните установихме още, че не всички атрибути характеризират достатъчно добре всеки един клас вино. Имаше доста атрибути, които почти не се различаваха по стойности за различните степени на качество. Имаше и такива, при които се наблюдават екстремни стойности (outliers).

### Червено вино (Атрибут ~ Качество)



### Бяло вино (Атрибут ~ Качество)



## Реализация

За програмната реализация използвахме R Project. Използвахме следните два алгоритъма:

* Random forest ([R имплементация](https://cran.r-project.org/web/packages/randomForest/randomForest.pdf))
  + Направихме експерименти с различен брой дървета в интервала [10,100]
* Neural Networks ([R имплементация](https://cran.r-project.org/web/packages/RSNNS/RSNNS.pdf))
  + Използвахме персептрон с един скрит слой. Броят неврони в скрития слой е 100. Броят максимални итерации за обучение е 1000. За първоначални тегла са използвани случайно избрани. Обучаващата функция е backpropagation.
  + Направихме експерименти с различна стойност на параметъра на функцията за обучение, като използвахме стойности в интервала [0,1] с промяна в стотните.

# Резултати от експерименти

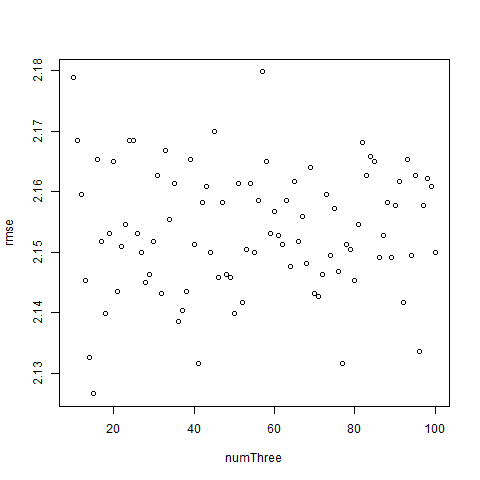
Проведохме множество от експерименти, като ги разделихме на няколко основни типа в зависимост от начина на обучаване и данните, върху които се изпробва модела. Проведохме същите типове експерименти както с *random forest,* така и с *neural networks* подхода. Ето и резултатите:

## Random forest резултати

### Обучаване с бяло вино(всички атрибути) тестване с бяло вино

Precision:

3 - 0.0000000

4 - 0.6000000

5 - 0.6830986

6 - 0.6310345

7 - 0.7258065

8 - 0.7500000

9 - 0.0000000

Recall:

3 - 0.0000000

4 - 0.1578947

5 - 0.6736111

6 - 0.8061674

7 - 0.4687500

8 - 0.4736842

9 - 0.0000000

F1:

3 - 0.0000000

4 - 0.2500000

5 - 0.6783217

6 - 0.7079304

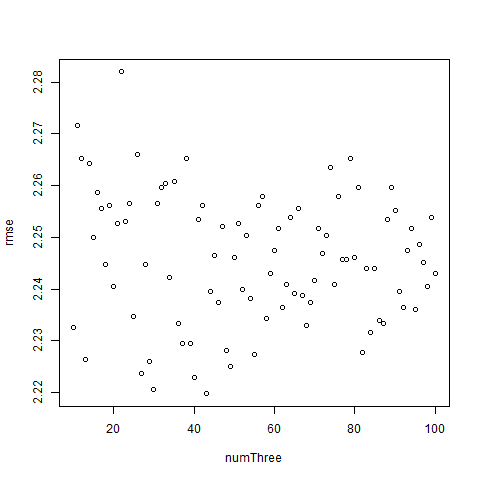
7 - 0.5696203

8 - 0.5806452

9 - 0.0000000

### Обучаване с бяло вино(избрани атрибути) тестване с бяло вино

Precision:

3 - 0.0000000

4 - 0.0000000

5 - 0.4827586

6 - 0.5153584

7 - 0.5000000

8 - 0.4285714

9 - 0.0000000

Recall:

3 - 0.0000000

4 - 0.0000000

5 - 0.4861111

6 - 0.6651982

7 - 0.3333333

8 - 0.1578947

9 - 0.0000000

F1:

3 - 0.0000000

4 - 0.0000000

5 - 0.4844291

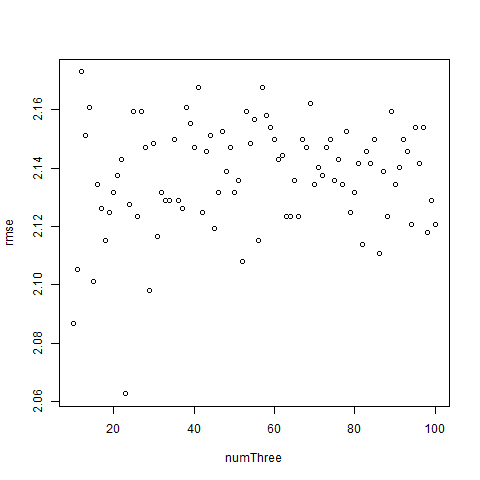
6 - 0.5807692

7 - 0.4000000

8 - 0.2307692

9 - 0.0000000

### Обучаване с червено вино(всички атрибути) тестване с червено вино

precision

3 - 0.0000000

4 - 0.0000000

5 - 0.7954545

6 - 0.7384615

7 - 0.4285714

8 - 0.0000000

recall

3 - 0.0000000

4 - 0.0000000

5 - 0.8641975

6 - 0.7058824

7 - 0.4000000

8- 0.0000000

f1

3 - 0.0000000

4 - 0.0000000

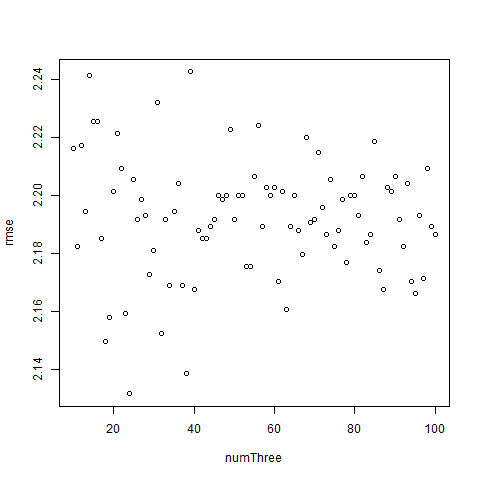
5- 0.8284024

6 - 0.7218045

7 - 0.4137931

8 - 0.0000000

### Обучаване с червено вино(избрани атрибути) тестване с червено вино

 Precision:

3 - 0.0000000

4 - 0.0000000

5 - 0.6813187

6 - 0.5932203

7 - 0.4375000

8 - 0.0000000

Recall:

3 - 0.0000000

4 - 0.0000000

5 - 0.7654321

6 - 0.5147059

7 - 0.4666667

8 - 0.0000000

F1:

3 - 0.0000000

4 - 0.0000000

5 - 0.7209302

6 - 0.5511811

7 - 0.4516129

8 - 0.0000000

### Смесено: обучаване с бяло вино(всички атрибути) тестване с червено вино

RMSE: 2.784196

Precision:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.00000000 | 0.05904762 | 0.50664137 | 0.52162162 | 0.25000000 | 0.00000000 | 0.00000000 |

Recall:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.00000000 | 0.60784314 | 0.44500000 | 0.33859649 | 0.01086957 | 0.00000000 | 0.00000000 |

F1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.00000000 | 0.10763889 | 0.47382431 | 0.41063830 | 0.02083333 | 0.00000000 | 0.00000000 |

### Смесено: обучаване с червено вино(всички атрибути) тестване с бяло вино

RMSE: 2.439979

Precision:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.00000000 | 0.07692308 | 0.44901610 | 0.46689895 | 0.31589744 | 0.00000000 | 0.00000000 |

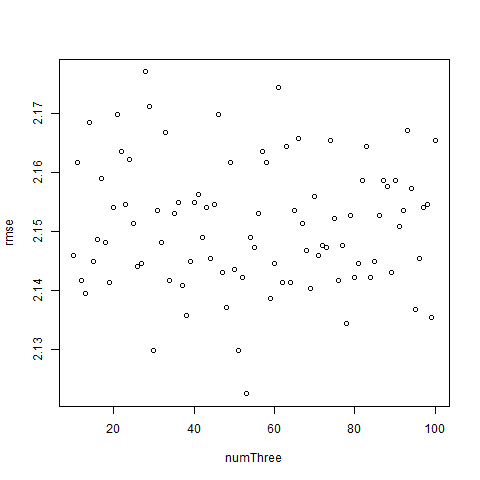
Recall:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.0000000 | 0.0069444 | 0.5734958 | 0.4079147 | 0.3928571 | 0.0000000 | 0.00000000 |

F1:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0.00000000 | 0.01273885 | 0.50367893 | 0.43541836 | 0.35019898 | 0.00000000 | 0.00000000 |

### Смесено: обучаване с червено и бяло вино(всички атрибути) тестване с бяло вино



## Neural Networks резултати

Резултатите имат следния вид: Първо е записан класът към който е класифициран примера после метриката за него.

### Обучаване с бяло вино(всички атрибути) тестване с бяло вино

maxAccuracy: 0.446183953033268 , l = 0.01

"5" - f1: 0.013793103448275, precision: 0.00694444444444444, recall: 1

"6" - f1: 0.616010854816825, precision: 1, recall: 0.445098039215686

### Обучаване с червено вино(всички атрибути) тестване с червено вино

maxAccuracy: 0.402366863905325 , l = 0.14

"5" - f1: 0.573839662447257, precision: 1, recall: 0.402366863905325

### Смесено: обучаване с червено и бяло вино(всички атрибути) тестване с бяло вино

maxSum: 1.44422700587084 , l = 0.05

recall: 1 accuracy; 0.444227005870841 , l = 0.05

"5" - f1: 0.43969465648855, precision: 1, recall: 0.281800391389432

### Смесено: обучаване с червено и бяло вино(избрани атрибути) тестване с бяло вино

maxSum: 1.44422700587084 , l = 0.05

recall: 1 accuracy; 0.444227005870841 , l = 0.05

"5" - f1: 0.43969465648855, precision: 1, recall: 0.281800391389432

# Заключение и бъдещо развитие

Нa базата на проведените експерименти и резултатите от тях можем да заключим, че използваните алгоритми не са най-подходящите за поставения проблем. Като цяло разпределението на данните е такова, че има множество примери за средния клас, но липсват примери за крайните класове, което затруднява алгоритмите. Експериментирането, както с промяна на параметрите, така и с промяна на множеството обучаващи примери, не довежда до значителна промяна в резултатите.

Като бъдещо развитие може да се експериментира с други алгоритми като например Наивен Бейсов класификатор. Друга възможност е да се направи предварителна обработка, която разделя качеството на виното само на 3 класа : „добро“, „нормално“, „лошо“ и отново да се експериментира с различни алгоритми.

# Разпределение на задачите

Сияна Славова пое реализацията на проблема с неврони мрежи, докато Иван Капукаранов се съсредоточи върху решението с дървета. Експериментите се проведоха взаимно на различни работни станции.

# Литература и използвани източници

* Wine dataset:
  + <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/>
* Съществуващо решение:
  + <http://www.r-bloggers.com/predicting-wine-quality-using-random-forests/>
* Random forest algorithm:
  + <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>
* RSNNS – neural networks algorithm
  + <https://cran.r-project.org/web/packages/RSNNS/RSNNS.pdf>
  + <https://www.jstatsoft.org/article/view/v046i07>
* R Project
  + <https://www.r-project.org/>