УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП

ФАКУЛТЕТ ЗА ИНФОРМАТИКА

ШТИП

Љупче Јаневски

АНАЛИЗА НА ПРИСТАПОТ КОН Е-УЧЕЊЕ СО ТЕХНИКИ ЗА ОБРАБОТКА НА ГОЛЕМИ ПОДАТОЦИ ОД MOODLE БАЗА НА ПОДАТОЦИ

-МАГИСТЕРСКИ ТРУД-

Штип, ноември, 2019 година

Комисија за оценка и одбрана:

Претседател:	**проф. д-р Цвета Мартиновска-Банде** **Факултет за информатика** **Универзитет „Гоце Делчев“ Штип**
Ментор:	**проф. д-р Зоран Здравев** **Факултет за информатика** **Универзитет „Гоце Делчев“ Штип**
Член:	**проф. д-р Александар Крстев** **Факултет за информатика** **Универзитет „Гоце Делчев“ Штип**

Датум на одбрана: _____

Датум на промоција: _____

Рецензирани и објавени трудови

1. Ljupce Janevski, Aleksandar Velinov, Zoran Zdravev (2019). **ANALYZING TEACHERS BEHAVIOR USING MOODLE DATA AND BIG DATA TOOLS**. *BJAMI, year2019, volume II, number1.*

Краток извадок

Денес постојано се генерира голем број на податоци кои бараат соодветна обработка. Стандардните техники за обработка не можеа да одговорат на овие барања. Развојот на информатичката технологија допринесе за појава на области чија примарна цел е обработка и анализа на податоците. Големите податоци се податоци кои се појавуваат со голема брзина, имаат голема количина и можат да бидат различен формат.

Поради појава на големите податоци се промени и начинот на обработка на податоците и поради тоа традиционалните алатки за обработка на податоци не се соодветни за процесирањето на податоци и се развиле алатки кои овозможуваат обработка и анализа на големи податоци. Анализата на големи податоци претставува процес на обработка на огромни податочни множества, со цел откривање на скриени или непознати релации и извлекување на значајни информации. Со ваква обработка на податоците може да се лоцираат одредени проблеми или недостатоци во текот на работењето кои не би можеле да се утврдат на друг начин.

Оваа магистерска работа се состои во истражување на техниките за анализа и обработка на големи податоци со цел добивање на корисни информации од огромната количина на необработени податоци. Притоа, целта е да се направи обработка на податоци генерирани од систем за електронско учење - Moodle со примена на алатките за процесирање на големи податоци како и со примена на некоја од техниките за податочно рударење. Врз основа на добиените резултати е важно да се пронајдат и дефинираат одредени особини за начинот на користење на системот за електронско учење, како и да се детерминираат неговите предности и недостатоци. На овој начин, со вакви сознанија, се гради основа за добивање нови идеи и методи за подобрување, односно оптимизирање на процесите на учење.

Клучни зборови: големи податоци, податочно рударење, Moodle, систем за електронско учење, кластерирање

Abstract

Today in almost every company almost can't be imagined without the use of hardware and software tools at work. As a consequence of the development of information technology, a large number of data to be processed has resulted. Standard processing techniques can't meet these requirements, therefore a new approach to the analysis and processing of these large data has emerged. Large data are data that appear at high speed, have a large amount and can be different format.

Due to the occurrence of large data has changed the way data processing and therefore traditional data processing tools are not suitable for data processing and tools have been developed that enable the processing and analysis of large data. Large data analysis is a process of processing huge data sets in order to detect hidden relationships or unknown connections and extract significant information. Such processing of data can identify some problems or shortcomings in the course of work that could not be determined otherwise.

In this master tesis consists in researching the techniques for analyzing and processing large data in order to obtain useful information from a huge amount of raw data. In doing so, the goal is to make data processing generated by the e-learning system - Moodle using the tools for processing large data as well as by applying one of the data mining techniques. Based on the results obtained, it is important to find and define certain characteristics of the way in which the e-learning system is used, as well as to determine its advantages and weaknesses. In this way, with such knowledge, the basis for obtaining new and ideas for methods of improvement, that is optimization of learning processes.

Keywords: Big Data, data mining, Moodle, e-learning systems, clustering

Содржина

1. Вовед 10
2. Цел на истражување 12
3. Дефиниција на поимот „Големи податоци“ 14
3.1 Управување со големи податоци 16
4. Машинско учење 18
4.1 Учење во длабочина 19
4.2 Поединечно и групно учење 21
4.3 Грануларно пресметување 21
4.4 Алгоритми за машинско учење 24
4.4.1 Надгледувано учење (Supervised learning) 24
4.4.1.1 Класификација (Classification) 24
4.4.1.1.1 Дрво на одлучување (Decision tree) 25
4.4.1.1.2 К најблиски соседи (K nearest neighbors) 26
4.4.1.2 Регресија (Regression) 27
4.4.1.2.1 Линеарна регресија (Linear regression) 28
4.4.1.2.2 Логичка регресија (Logistic regression) 29
4.4.2 Учење без надзор (Unsupervised learning) 29
4.4.2.1 Кластерирање(Clustering) 30
4.4.2.1.1 K-means 31
4.4.2.2 Асоцијација(Association) 32
5. Алатки за анализа и обработка на големи податоци 34
5.1 Hadoop 34
5.1.1 Ниво на складирање на податоци 35
5.1.1.1 Hadoop дистрибуиран систем на датотеки (HDFS) 36
5.1.1.2 HBase 38
5.1.2 Ниво на обработка на податоци 39
5.1.2.1 Map Reduce 39
5.1.2.2 YARN 40
5.1.3 Ниво на пристап до податоци 41
5.1.3.1 Pig 41
5.1.3.2 Hive 42
5.1.3.3 JAQL 43
5.1.3.4 Sqoop 43
5.1.3.5 Mahout 44
5.1.3.6 Flume 45

5.1.3.7 Chukwa 45
5.1.4 Ниво на управување со податоци 46
5.1.4.1 Oozie 46
5.1.4.2 Ambari 47
5.1.4.3 Whirr 47
5.1.4.4 BigTop 48
5.1.4.5 Hue 48
5.1.4.6 ZooKeeper 48
5.1.4.7 Avro 49
5.2 Дистрибуции на Hadoop 49
5.2.1 Cloudera 50
5.2.2 MapR 51
5.2.3 Hortonworks 51
6. Истражување 52
6.1 Moodle систем 52
6.2 Фази на истражувачката работа 53
6.2.1 Собирање на податоци 53
6.2.2 Препроцесирање и трансформација на собраните податоци и креирање на податочен сет погоден за понатамошна обработка 55
6.2.3 Примена на алгоритми за податочно рударење (Кластерирање со K-means) 60
6.2.4 Евалуација и анализа на добиените резултати 68
7. Заклучок 77
8. Користена литература 80

Слики

СЛИКА 1. ВОДОПАДЕН МОДЕЛ ЗА ЧЕКОРИТЕ ВО ИСТРАЖУВАЧКАТА РАБОТА 13
СЛИКА 2. ДРВО НА ОДЛУЧУВАЊЕ..... 26
СЛИКА 3. ПРИМЕР ЗА ЛИНЕАРНА РЕГРЕСИЈА 28
СЛИКА 4. КОМПОНЕНТИ НА APACHE HADOOP..... 35
СЛИКА 5. АРХИТЕКТУРА НА HDFS И MAPREDUCE НИВОТО 37
СЛИКА 6. HUE-ДЕЛОТ ОД HADOOP СО ИМПОРТИРАНИТЕ ТАБЕЛИ ОД MOODLE БАЗАТА НА ПОДАТОЦИ..... 56
СЛИКА 7. ДОБИВАЊЕ НА ТАБЕЛА НА НАСТАВНИЦИ 56
СЛИКА 8. БЛОК-ДИЈАГРАМ ЗА ДОБИВАЊЕ НА СИТЕ АКТИВНОСТИ НА НАСТАВНИЦИТЕ..... 57
СЛИКА 9. RAPIDMINER STUDIO 61
СЛИКА 10. ПРИМЕНА НА ELBOW МЕТОД ПРИ ИЗБОР НА БРОЈОТ НА КЛАСТЕРИ К..... 62
СЛИКА 11. МАТРИЦА НА КОРЕЛАЦИЈА НА АТРИБУТИТЕ..... 63
СЛИКА 12. НОРМАЛИЗИРАН ПОДАТОЧЕН СЕТ..... 64
СЛИКА 13. БРОЈ НА НАСТАВНИЦИ ВО СЕКОЈ ОД ДОБИЕНИТЕ КЛАСТЕРИ..... 65
СЛИКА 14. ГРАФИКОН НА РАСТОЈАНИЕ НА АТРИБУТИТЕ ОД ЦЕНТАРОТ НА ЦЕНТРОИДИТЕ .. 66
СЛИКА 15. ДРВО НА ОДЛУЧУВАЊЕ..... 66
СЛИКА 16. ГРАФИКОН НА АТРИБУТИТЕ ВО КЛАСТЕРИТЕ..... 67
СЛИКА 17. ПРИКАЗ НА КОРЕЛАЦИЈАТА МЕЃУ МОДУЛИТЕ: ИЗБОР, КНИГА, ЗАДАЧА, ЛЕКЦИЈА И СООДВЕТНИ КЛАСТЕРИ..... 70
СЛИКА 18. РАСПРЕДЕЛБА НА НАСТАВНИЦИТЕ ВО ЗАВИСНОСТ ОД БРОЈОТ НА АКТИВНОСТИ ВО МОДУЛОТ ЗАДАЧА (ASSIGN) 71
СЛИКА 19. РАСПРЕДЕЛБА НА НАСТАВНИЦИТЕ ВО КЛАСТЕРИТЕ ВО ЗАВИСНОСТ ОД БРОЈОТ НА АКТИВНОСТИ ВО МОДУЛОТ КНИГА (BOOK) И РАЗГОВОР (CHAT) 72
СЛИКА 20. РАСПРЕДЕЛБА НА НАСТАВНИЦИТЕ ВО КЛАСТЕРИТЕ ВО ЗАВИСНОСТ ОД БРОЈОТ НА АКТИВНОСТИ ВО МОДУЛОТ ИЗБОР (CHOICE) И ФОРУМ (FORUM)..... 73
СЛИКА 21. РАСПРЕДЕЛБА НА НАСТАВНИЦИТЕ ВО МОДУЛОТ РЕЧНИК (GLOSSARY) И ЛЕКЦИЈА (LESSON)..... 74

СЛИКА 22. РАСПРЕДЕЛБА НА НАСТАВНИЦИТЕ ВО МОДУЛОТ КВИЗ (QUIZ)..... 75

Табели

ТАБЕЛА 1. ПРИМЕР ЗА АСОЦИЈАТИВНО ПРАВИЛО ВО СУПЕРМАРКЕТОТ . **ERROR! BOOKMARK NOT DEFINED.**

ТАБЕЛА 2. ВКУПНИ АКТИВНОСТИ НА НАСТАВНИЦИТЕ.....55

1. Вовед

Денес со развојот на информатичката технологија и Интернетот, дојде до зголемувањето на количината на податоци кои се создаваат во структуриран и неструктуриран облик, па затоа овие таканаречени големи податоци честопати станаа предмет на анализа на речиси сите непрофитни и профитни организации. Овие податоци, добиени со информациските решенија и системи се причина за постигнување успех во работењето како и остварување предност пред конкуренцијата на компанииите. Од овие причини довело до појава на нов концепт таканаречен „Големи подататоци“ (Big Data) и развој на нови технологии и алатки за нивна обработка и анализа. Така што во првиот дел на оваа магистерска работа се опишани фундаменталните карактеристики на овие податоци.

Поради комплексноста на големите податоци, обработката на големите податоци не е возможна да се прави со традиционалните алатки и технологии, затоа дошло до појава на нови решенија , кои можат да одговорат на барањето за процесирање на податоци со голема брзина, разновидност во форматот и структурата на големите податоци. Алатките кои се користат за анализа на големи податоци се презентирани во делот „Алатки за анализа на големи податоци“.

Во третиот дел „Анализа на големи податоци со алгоритми за машинско учење“ претставени се видовите машинско учење како и некои од најчесто применуваните алгоритми за машинско учење. Во овој дел се објаснети најзначајните особености на овие алгоритми, а подетално е прикажан K-means алгоритамот како еден од најкористените алгоритми за кластерирање (техника за учење без надзор).

Предмет на анализа во овој магистерски труд се податоци добиени од Moodle база на податоци на Универзитетот „Гоце Делчев” – Штип. Во делот „Едукативно податочно рударење“ се објаснети главните особини на големи податоци генерирани од системи кои се користат во образованието и кои се техниките кои се применуваат со цел извлекување на колку што е можно посуштински информации од огромните волумени податоци и создавање на одредено знаење врз основа на пронајдените релации и логичности.

Идејата на оваа магистерска работа се состои во анализа на податоци добиени од систем за електронско учење – Moodle, при што врз основа на видот на податоците во базата на податоци, да се извлечат одредени битни информации, односно да се екстрахира знаење од масивното податочно множество. Притоа, фокусот е ставен на процесирање на активностите на наставниците како корисници на системот за електронско учење. Примарната цел пак, на магистерската работа е да се направи дескриптивна анализа на употребата на системот за електронско учење Moodle во одреден временски период и врз основа на таа анализа да се донесат соодветни заклучоци во врска со користењето на одредени модули од системот. Врз основа на добиените резултати од обработката на податоците се очекува да се добијат и одредени сознанија за едукативниот процес, неговите предности и недостатоци. Како резултат на тоа, би се овозможило детерминирање на слабите страни во употребата на системот за електронско учење, а со тоа и на дел од образовниот процес.На овој начин, преку утврдување на слабостите се отвораат можности за наоѓање методи за нивно отстранување или минимизирање на ефектите од нив врз крајниот исход во едукативните процеси.

Во делот „Истражување“, се презентирани фазите од истражувачката работа, користените алатки, добиените резултати, како и дел од имлементираните програмски кодови за извршување на анализата. Преку претставените графикони и табели се дадени секој од чекорите во фазите од истражувањето како и исходот од секој од реазлизираните чекори. Во овој дел, исто така, е направена анализа на добиените резултати од примената на алатките, и е дадена соодветна интерпретација за добиениот исход. Дополнително, се претставени предлози за тоа како добиените резултати, односно како овие техники за анализа би можеле да најдат примена за подобрување на процесите на учење и нивна оптимизација.

Во делот „Заклучок“ се наведени финални толкувања за реализираната истражувачка работа, направена е паралела помеѓу почетните цели на работата и добиените крајни резултати. Дополнително, во овој дел се претставени искуствата од употребените техники за обработка и е направен преглед на исходот од истражувањето. Понатаму, во завршниот дел од магистерската теза, е дадено

размислување за тоа како добиените резултати наоѓаат примена во утврдување на слабите страни во образовниот процес. Ова се прави со единствена цел максимално искористување на капацитетите на актерите во едукативните процеси и на можностите што ги нудат новите информатички технологии за подобрување на образовните системи во целина.

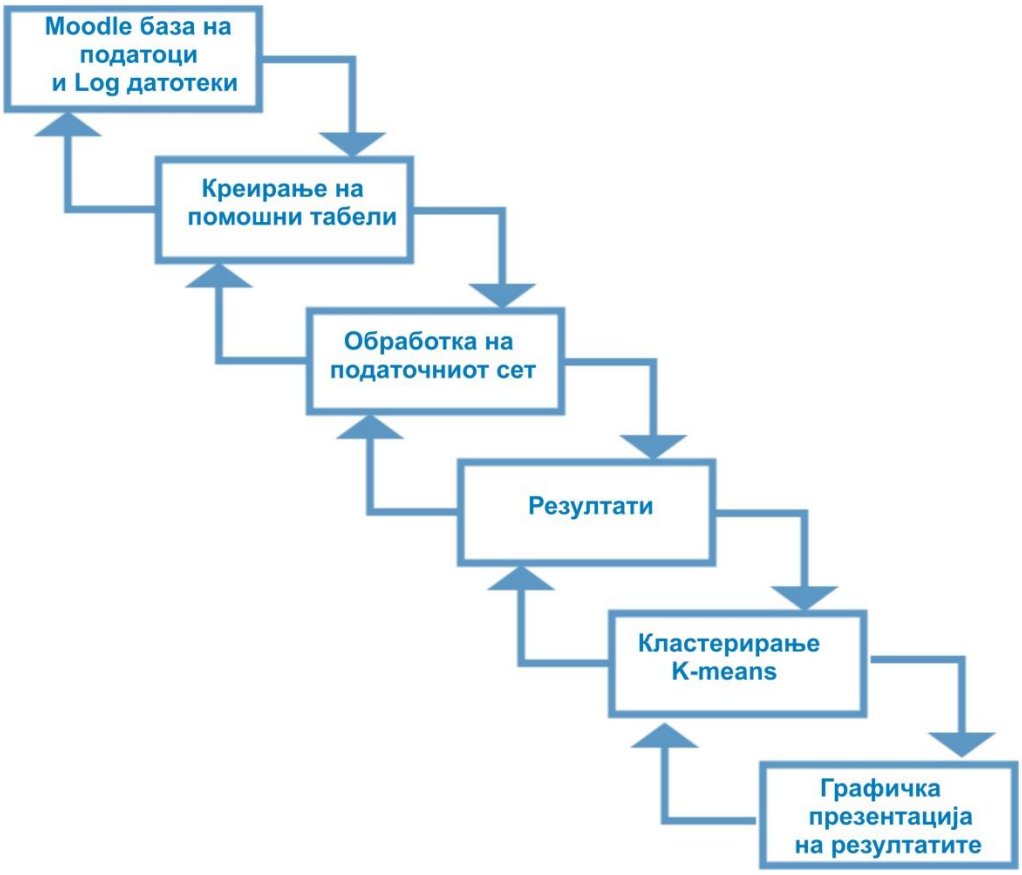
2. Цел на истражување

Главна цел на овој труд е анализа на големи податоци добиени од Moodle база на податоци на Универзитетот „Гоце Делчев" – Штип. Базата на податоци содржи податоци за подолг временски период од 2012 до 2019 година и нејзината големина изнесува 13 GB. Станува збор за огромна количина податоци. Поради тоа, е потребно да се користат посебни технологии и алатки, наменети за процесирање на големи податоци. Со цел да се олесни работата за добивање на активноста на наставниците во Moodle, создадовме помошни табели. Овие табели ги содржат податоците за индивидуалните корисници за одредена активност на некој модул. Ние ги истражувавме активностите на наставниците за разни модули на Moodle, како што се форуми, време, речник, задачи, анкети, квизови, избори, чатови, вики и книги. Како најпогоден начин за обработка е избрана техниката за податочно рударење – кластерирање, поточно примена на алгоритамот за машинско учење – K-means. Основната цел е да се направи дескриптивна анализа на начинот на тоа како наставниците го користеле системот за електронско учење, во наведениот период. Основни задачи кои треба да се реализираат се:

- Селекција на потребните податоци од базата на податоци со цел создавање на добар податочен сет за понатамошна анализа;
- Имплементација на алгоритам за кластерирање;
- Евалуација и интерпретирање на резултатите.

Генерално гледано, целта е да се добијат информации за користењето на системот за е-учење на Универзитетот „Гоце Делчев“ - Штип. Со добивање на заклучоци врз основа на добиените резултати може да се идентификуваат слабостите и недостатоците при користење на системот за образовни цели. Врз основа на тоа, е потребно да се пронајдат начини и да се утврдат методи за надминување на слабостите и подобрување на едукативниот процес.

Планот за имплементирање на ова истражување, заради постигнување на наведените цели, е прикажан со водопаден модел претставен на Слика 1.



Слика 1 Водопаден модел за чекорите во истражувачката работа

Figure 1 Waterfall model for the steps in research work

Со овој модел на водопад се анализира однесувањето на наставниците кои користат големи податоци од базата на податоци на Moodle и алатките за големи податоци. Главната цел на презентираниот модел е да се извлече знаење од активностите на наставниците. Во иднина можеме да го искористиме стекнатото

знаење за да го подобриме наставниот процес и да создадеме нови образовни методи.

3. Дефиниција на поимот „Големи податоци“

Терминот „Големи податоци“ се однесува на било кој сет на податоци [1] кој е толку голем или толку сложен, така што конвенционалните апликации не се соодветни за да го процесираат. Терминот исто така се однесува на алатките и технологиите кои се користат за да се справат со големите податоци. Примери за големи податоци вклучуваат количина на податоци споделени на Интернет секој ден, видеа на YouTube, Твитер постови и податоци за локацијата на мобилниот телефон. Во последниве години, податоците произведени од институции за учење, исто така, почнаа да стануваат доволно големи за да ја зголемат потребата за технологии и алатки за анализа на големи податоци.

Големите податоци не се единствена технологија, туку комбинација на стари и нови технологии што им помагаат на компаниите да добијат вистински увид. Затоа, големите податоци ја имаат способноста да управуваат со огромен волумен на различни податоци, со вистинска брзина и во вистинскиот временски период за да овозможат анализа и реакција во реално време. Четири особини се карактеристични за големите податоци [2] :

• Количина - колку податоци,
• Брзина - колку брзо се обработуваат овие податоци,
• Веројатност - квалитетот на достапните податоци и
• Разновидност - различни типови на податоци.

Големите податоци може да се анализираат за да се најдат асоцијации, модели, трендови за да се добие знаење од нив.

Големите податоци [3] може да се дефинираат како средина која содржи алатки, процеси и процедури кои го поттикнуваат откривањето на податоците во неговиот центар. Овој процес на откривање се однесува на нашата способност да се извлече деловната вредност од податоците и вклучува собирање, манипулирање, анализирање и управување со податоците.

Станува збор за четири дискретни својства на големите податоци кои бараат специјални алатки, процеси и постапки за да се справат со:

• Зголемени количини (до степен на петабајти и повеќе),
• Зголемена достапност / достапност на податоци (во реално време),
• Зголемени формати (различни типови на податоци) и
• Зголемена мешавина (густа распределба).

Поради зголемениот обем на податоци, тие не можат да се анализираат со стандардните техники за обработка на бази на податоци. За ова се потребни специјални алатки за анализа на големи податоци [4] кои имаат имплементирано разни методи за побрза обработка и прибирање на резултатите. Овие податоци се достапни и анализата во реално време може да обезбеди знаење кое може да го подобри процесот на учење во случај на образовни податоци, на пример. Овие податоци може да се појават во различни формати. Некои од нив се структурирани, додека други се полуструктурирани и неструктурирани. Методот на анализа на структурирани податоци е многу полесен во споредба со неструктурирани податоци и полуструктурирани податоци. Често во нашата анализа треба да комбинираме повеќе од овие типови на податоци со цел да добиеме подобри резултати.

Особено интересна област се големите податоци во образованието. Овие податоци често доаѓаат од системите за управување со учењето (LMS). Со анализирање на овие податоци, можеме да извлечеме знаење кое може да ни помогне да го подобриме процесот на учење, да ги подобриме курсевите и начинот на кој се организираат курсевите. На овој начин можеме да утврдиме што е добро или што е лошо и со тоа можеме да го промениме образовниот процес. Ова е многу корисно за наставниците. Користејќи анализа на големи податоци, наставниците можат да откријат како наставните методи влијаат врз однесувањето на учениците. Интересни информации кои можат да се добијат од анализата се: број на преземени ресурси, број на поднесени задачи, активност на учениците на курсеви и сл. Ова е добар фидбек за наставниците за нивната работа. Наставниците можат да откријат нови методи користејќи добри практики кои придонесуваат за поголема активност и подобар студентски успех [5]. Според

студентските активности (на пример, домашните задачи), можеме да откриеме кои ресурси за учење се навистина добри за совладување на материјалот и кои ресурси треба да се подобрат, надградат или да се додадат нови ресурси. На овој начин, педагошките пристапи може да се земат во предвид при учењето на курсот.

3.1 Управување со големи податоци

Рударењето на големи податоци нуди многу атрактивни можности. Сепак, истражувачите и професионалците се соочуваат со неколку предизвици при истражување на множествата на големи податоци и кога добиваат вредности и знаење, добиваат малку информации на различни нивоа, вклучувајќи: собирање, складирање, пребарување, споделување, анализа, управување и визуелизација. Понатаму, постојат прашања за безбедноста и приватноста, особено во дистрибуирани податочни управувани апликации. Честопати, потопувањето на информациите и погрешно насочени правци ја надминува нашата способност за искористување на големите податоци. Всушност, иако големината на големите податоци постојано се зголемува експоненцијално, сегашниот технолошки капацитет на нас е да се справиме со множествата на големи податоци со големини од petabytes, exabytes и zettabytes. Научниците се соочуваат со многу предизвици кога се занимаваат со големи податоци. Еден предизвик е како да се соберат, интегрираат и складираат огромни збирки на податоци генерирани од дистрибуирани извори [6] со помалку барања за хардвер и софтвер. Друг предизвик е управување со големи податоци, сигурен резултат и да се оптимизираат трошоците. Всушност, доброто управување со податоци е темелот за анализа на големи податоци. Големи податоци за управување значи да се исчистат податоците за сигурност, да се агрегираат податоците кои доаѓаат од различни извори и да ги кодираат заради нивна безбедност и приватност. Со други зборови, целта за управување со големи податоци е да се обезбедат доверливи податоци кои се лесно достапни, управливи, соодветно зачувани и обезбедени. Постојат 5 чекори што треба да се направат, пред да се изврши управување со податоци (чистење, агрегација, кодирање, складирање и пристап).

Предизвикот кај големите податоци, претставува справување со комплексната природа на големите податоци (брзина, обем и разновидност) [7] и обработката во дистрибуирана средина со мешавина на различни апликации. Всушност, за сигурни резултати од анализата, е неопходно да се потврди веродостојноста на изворите и квалитетот на податоците пред ангажирањето на ресурсите. Сепак, изворите на податоци може да содржат звуци, грешки или нецелосни податоци. Предизвикот е како да се исчистат таквите огромни збирки на податоци и како да се одлучи за тоа кои податоци се сигурни и кои податоци се корисни.

На пример, анализата на податоци ѝ овозможува на организацијата да извлече вреден увид и да ги следи моделите кои можат да влијаат позитивно или негативно на бизнисот. Другите апликации за управување на податоци, исто така, имаат потреба од анализи во реално време: како навигација, социјални мрежи, финансии, биомедицина, астрономијата, интелигентни транспортни системи. Така, се потребни напредни алгоритми и практични методи за рударење на податоци за да се добијат точни резултати, да се следат промените во различни полиња и да се предвидат идните набљудувања. Сепак, анализата на големи податоци сè уште е предизвикувачка поради многу причини: комплексната природа на големите податоци, вклучувајќи ги и петте чекори при анализа, потребата за приспособливост и перформанси за анализирање на такви хетерогени множества на податоци со реална реакција [8].

Денес, постојат различни аналитички техники, вклучувајќи податоци за рударство, визуелизација, статистичка анализа и машинско учење. Многу студии се справуваат со оваа област со помош на подобрување на употребените техники, предлагање на нови или тестирање на комбинација на важни алгоритми и технологии. Така, големите податоци го туркаа развојот на системските архитектури, хардверот, како и софтверот. Сепак, сè уште ни е потребен аналитички напредок за да се соочиме со предизвиците за големи податоци и обработка. Едно од прашањата е како да се гарантира навременоста на одговорот кога обемот на податоци е многу голем? Во следните делови, ги истражуваме испитувањата на разликите со кои се соочуваме при примената на тековните

аналитички решенија: машинско учење, длабоко учење, поединечно учење, грануларни пресметки.

4. Машинско учење

Целта на машинското учење е да открие знаење и да се донесат интелигентни одлуки. Се користи за многу реални апликации како што се мотори за препораки, системи за препознавање, информатичка технологија и податоци за рударство, како и автономни системи за контрола. Генерално, машинското учење (ML) е поделено на три поддомени: надгледувано учење, учење без надзор и засилено учење.

Тековните апликации од реалниот свет, како што се сензорските мрежи, трансакциите со кредитни картички, управувањето со акции, постови на блог произведуваат огромни сетови на податоци. Методите за рударење на податоци се важни за откривање интересни обрасци и екстракција на вредноста скриена во такви огромни сетови на податоци и потоци. Меѓутоа, традиционалните техники за рударство на податоци, како што се: асоцијација, рударство, кластерирање и класификација, имаме недостаток на ефикасност, скалабилност и точност кога се применуваат на таквите големи податоци во динамична средина. Поради големината, брзината и варијабилноста на потоци, не е изводливо трајно да се чуваат, а потоа да се анализираат. Така истражувачите треба да дадат нови начини за оптимизирање на аналитичките технологии, да обработуваат податоци во многу ограничен временски период со ограничени ресурси (на пример меморија) и да произведуваат точни резултати во реално време. Понатаму, варијабилноста на потоци носи непредвидливи промени (промена на дистрибуција на инстанци) во дојдовните податоци. Затоа, неколку методи за податочно рударење беа прилагодени да вклучуваат техники за откривање на лебдење и да се справат со менување на животната средина. Класификација и кластерирање се најзначајните методи. Експериментите на податочните текови покажаа дека промените во основниот концепт влијаат на изведбата на моделот на

класификација. Така се потребни подобрени аналитички методи за да се откријат и да се прилагодат на концептот на лебдење [9].

Како пример во сегашната нестабилна економска средина, на претпријатијата им е потребен систем за предвидување на финансиски проблеми (FDP). Ваквиот систем е од суштинско значење за подобрување на управувањето со ризиците и поддршката на банките во кредитни одлуки. DFDP (Динамичка прогноза за финансиска криза) стана важна гранка на истражувањето на FDP [10]. Ги подобрува корпоративните управувачки ризици. Фокусот се става на тоа како динамично да се ажурира моделот FDP кога постепено се појавуваат нови податоци за примерокот.

4.1 Учење во длабочина

Денес, длабокото учење претставува исклучително активна област која се одржува во машинското учење и игра важна улога во апликативните аналитички апликации, како што се визуелизација на компјутер, препознавање на говор и обработка на природниот јазик [6].

Традиционалните техники за машинско учење и алгоритмите се ограничени во нивната способност да обработуваат природни податоци во нивната сирова форма [11]. Напротив, учењето во длабочина помага при решавање на аналитичките и проблемите во учењето кои се наоѓаат во огромни збирки на податоци. Тоа исто така помага при автоматско извлекување на сложени податочни репрезентации од големи количини на незаштитени и некатегоризирани сирови податоци. Бидејќи длабокото учење се заснова на хиерархиско учење и извлекување на различни нивоа на комплексни податоци, соодветно е да се поедностави анализата на големите количини на податоци со семантичко индексирање, означување на податоци, пронаоѓање информации и задачи. Овде спаѓаат класификација и предвидување (на пример екстрактор на функции што ги трансформира необработените податоци (како што се вредностите на пиксели на Сликата): во соодветна внатрешна репрезентација или вектор на карактеристики од кои потсистем за учење, честопати класификатор, може да ги открие или класифицира обрасците во влез. Сепак, и покрај овие

предности, големите податоци сè уште претставуваат значителан предизвик за длабоко учење [12]:

- огромни количини на големи податоци: фазата на обука не е лесна задача за учење на големи податоци воопшто и за длабоко учење. Ова е затоа што итеративните пресметки на алгоритмите за учење се многу неспоредливи за паралелизирање. Така, сè уште постои потреба да се создадат ефективни и скалабилни паралелни алгоритми за да се подобри фазата на обука за овој модел;
- хетерогеност: големите количини на податоци наметнуваат голем предизвик за длабоко учење. Тоа значи да се справи со голем број примери (влезови), големи сорти на класни типови (излези) и многу висока димензионалност (атрибути). Така, аналитичките решенија треба да се справат со сложеноста на комплексноста и комплексноста на моделот. Освен тоа, таквите големи количини на податоци го прават неизводливо обучувањето на алгоритам за длабоко учење со централен процесор и складирање;
- бучни етикети и нестационарна дистрибуција: поради различното потекло и хетерогени извори на големи податоци, аналитичките истражувачи сè уште се соочуваат со други предизвици, како што се непотполноста на податоците, недостасува ознаки и бучни етикети;
- голема брзина: како што знаеме, податоците се генерираат со екстремно голема брзина и треба да се обработуваат во реално време. Покрај високата брзина, податоците честопати се нестационарни и со текот на времето ја менуваат дистрибуцијата.

Поради овие цитирани прашања, решенијата за длабоко учење сè уште немаат зрелост и им треба дополнително детално истражување за да ги оптимизираат аналитичките резултати. Во краток преглед, идните истражувања, треба да се разберат како да се подобрат алгоритмите за длабоко учење, со цел да се справат со анализата на стриминг податоци, високата димензионалност и приспособливост на моделите.

4.2 Поединечно и групно учење

Поединечното учење и учењето на групи претставуваат две фундаментални методи за учење од големи потоци податоци со концепт лебдење [13].
Поединечното и групното учење често се применуваат на големите податоци и се справуваат со разни побарувања, како што се решавањето на достапноста на податоците и ограничените ресурси. Тие се прилагодени на многу апликации како што се предвидување на тренд акции. Примената на инкременталното учење овозможува да се произведуваат побрзи временски периоди за време на приемот на нови податоци.

Многу традиционални алгоритми на машинско учење, инхерентно го подржуваат поединечното учење. Примери за поединечни алгоритми вклучуваат дрва на одлучување (IDE4, ID5R), правила на одлуки, невронски мрежи, невронски Гаусонови RBF мрежи (Learn ++, ARTMAP) или поединечни алгоритми. Од ова се забележува дека поединечните алгоритми се побрзи од традиционалните алгоритми. Сепак, групните алгоритми се пофлексибилни и можат подобро да се прилагодат на концептот лебдење. Покрај тоа, не можат да се користат сите класификациски алгоритам во постепеното учење, но речиси секој класификациски алгоритми може да се користи во групните алгоритми [13]. Така, се препорачува да се користи поединечен алгоритам во отсуство на концептот лебдење или ако концептот лебдење е мазен. Понатаму, ако треба да се справиме со релативно едноставен податочен поток или со високо ниво на процесирање во реално време, поприоритетното учење е посоодветно. Сепак, групното учење претставува подобар избор во случај на комплицирана или непозната дистрибуција на податочни текови.

4.3 Грануларно пресметување

Грануларното пресметување како модел не е нов, но неодамна стана популарен за неговата употреба во разни домени на големи податоци. Овој модел покажува многу предности во случај на анализа на интелигентни податоци,

препознавање на модели, машинско учење и неизвесно реагирање за огромна големина на збирки на податоци. Овој модел одигра важна улога во дизајнот на моделите за донесување одлуки, притоа обезбедувајќи прифатливи перформанси. Технички, грануларното пресметување претставува општа теорија за пресметување врз основа на гранули, како што се класи, кластери, подгрупи, групи и интервали. [14] Така, тоа може да се искористи за изградба на ефикасен компјутерски модел за сложени апликации за големи податоци како што се податоци за рударство, анализа на документи, финансиски малверзации, организација и пронаоѓање на огромни бази на податоци за мултимедија, медицински податоци, далечинско сочувување, биометрика.

Дистрибуираните системи бараат поддршка на различни корисници во непостоечките големи податоци на различни нивоа на грануларност. Исто така, постои потреба да се анализираат податоците и да се презентираат резултатите со различни ставови. За да ги исполни овие барања, овој модел обезбедува моќни алатки за мулти-грануларност и повеќекратно гледање на анализата на податоците. Ова овозможува подобро разбирање и анализирање на сложеноста на различни големи множества на податоци. Покрај тоа, техниките можат да послужат како ефективни алатки за процесирање за реалните светски интелигентни системи и динамичко опкружување. Покрај тоа овозможува да се справи со сложеното прашање за развојот на атрибути и предмети со текот на времето. Навистина, овој модел одигра важна улога за да го приближи решението, истовремено обезбедувајќи ефикасност и подобрен опис. Овој модел може да се имплементира преку разни технологии, како што се: нејасни сетови, груби сетови, случајни множества итн. Општо земено, неопределените множества биле применети на различни области [15], како што се контролни системи, препознавање на модели и машинско учење. Нејасните множества ни овозможуваат да ги претставуваме и обработуваме информациите на различни нивоа на грануларност на информациите. Поспецифично, техниките на Fuzzy Set играат важна улога во сите фази на синџирот на вредности на големите податоци: Прво во справувањето со несигурностите на необработените податоци, а потоа ги

коментираат податоците и се подготвува специфична грануларна застапеност на податоци за вештачки интелигентни алгоритми.

На пример, Хуанг и Сор [16] докажаа дека нивниот застарен модел ги надминува статичните алгоритми и соодветните алгоритми за зголемување, како што е методот на Занг [13].Навистина, моделот обезбедува подобра ефикасност и оптимизира компартирање. За тоа, двете решенија на Хуанг и Сор, се засноваат на тие основи: прв метод на матрикс се користи за конструирање и пресметување на груби пристапи. Второ, се користи поединечен метод за динамичко ажурирање на приближувањето на груб сет.

Исто така забележуваме дека матриците сè повеќе се користат за груба анализа на податоци и приближување. Ова е затоа што структурата на матрицата поддржува опис на огромни збирки на податоци, одржливост и оптимизирани пресметки. Всушност, за да се извлече знаење од дојдовните потоци, моделот ги надградува и прави пресметки само на мали релативни матрици (подматрици), наместо да ја ажурира матрицата на целата врска. За разлика од оние пристапи базирани на матриксната операција, Луо [17] моделот користел веројатност на груб сет модел со инкременталниот пристап. Нивната цел е да моделираат непрецизни податоци со толеранција на грешки во одлучувањето во однос на условната веројатност и веројатните параметри.

Следствено, техниките на овој модел можат да ги подобрат сегашните техники за големи податоци, истовремено справувајќи се со големи предизвици за податоци (предизвици покренати од 5-те чекори, со претходна обработка на податоци или со реконструирање на проблемот на одредено грануларно ниво). Сепак, вреди да се забележи дека улогата на овој модел и нејасни сет техники е да се обезбеди методологија за апстракција на знаење (гранулација) и знаење на работната застапеност. Ова е различно од улогата на другите техники кои се користат за големи податоци, како што е длабоко учење [15].

4.4 Алгоритми за машинско учење

Машинското учење може да биде:
- надгледувано учење (Supervised learning) и
- учење без надзор (Unsupervised learning).

4.4.1 Надгледувано учење (Supervised learning)

Надгледувано учење е местото каде што имаме влезни променливи (x) и излезна променлива (Y) и користиме алгоритам за да ја научиме функцијата за мапирање од влезот до излезот.

Y = f (X)

Целта е да ја приближиме функцијата за мапирање, така што кога имаме нови влезни податоци (x) , толку добро може да ги предвидиме излезните променливи (Y) за тие податоци. Се нарекува надгледувано учење бидејќи процесот на учење на алгоритми од базата на податоци за обука го надгледува наставник. Ако ги знаеме точните одговори, алгоритмот итеративно прави предвидувања за податоците за обуката и ги корегира наставникот и учењето запира кога алгоритмот постигнува прифатливо ниво на перформанси. Кај ова учење, проблемите во учењето може понатаму да се групираат во проблеми со регресија и класификација.

-Класификација (Classification): Проблемот со класификација е кога излезната променлива е категорија(дискретна), како што се „црвена“ или „сина“ или „болест“ и „нема болест“.

-Регресија (Regression): Проблемот со регресијата е кога излезна променлива е вистинска вредност (број), како што се „долари“ или „тежина“.

4.4.1.1 Класификација (Classification)

Најпознати алгоритми кои ја користат техниката класификација овде се:
- дрво на одлучување (Decision tree) и
- К најблиски соседи (K nearest neighbors).

4.4.1.1.1 Дрво на одлучување (Decision tree)

Дрво на одлучување е систем за поддршка на одлуките кој користи одлуки во графикон сличен на дрво, вклучувајќи ги резултатите од случајните настани, трошоците за ресурси и алатки. Дрво на одлучување, или дрво на класификација, се користи за да се научи функцијата на класификација која ја прикажува вредноста на зависен атрибут (променлива) од вредностите на независни (влезни) атрибути (променливи). Овој проблем е познат како надгледувана класификација бидејќи зависниот атрибут и броењето на класите (вредности) се познати. Дрвото на одлучување е еден од најмоќните пристапи во откривањето на знаењето и податочното рударење. Тоа вклучува технологијата за истражување на големи и сложени податоци и е многу важно бидејќи овозможува моделирање и извлекување на знаење од најголемиот дел од податоците кои се на располагање. Сите теоретичари и специјалисти постојано бараат техники за да го направат процесот поефикасен, поекономичен и прецизен. Дрвата за одлучување се многу ефикасни алатки во многу области, како што се податоци и текстови за рударење, екстракција на информации, машинско учење и препознавање на модели. Дрвото на одлуки нуди многу придобивки за податоци од рударењето, како што се:

• Лесно е да се разбере од страна на крајниот корисник;

• Може да се справи со различни влезни податоци со номинална, нумеричка и текстуална вредност;

• Можност да се обработуваат податоци чии вредности се непознати;

• Високи перформанси;

• Овој алгоритам може да се имплементира преку различни платформи [10]. Дрвото на одлучувањето вклучува: јазол на одлучување, јазол настан и терминален јазол, како и гранки на одлучување и гранки настан. Крајниот резултат е дрво со добиени крајни јазли (наречени листови). Јазолот на одлука има две или повеќе гранки, а јазолот лист претставува донесена одлука, односно конечна класификација. Почетниот јазол на одлука се нарекува коренски јазол. Најпрво се започнува со избирање на највредниот атрибут кој се поставува како коренски јазол всушност го означува оној атрибут кој дава најмногу информации, или со

други зборови за кој пресметаната информациска добивка е најголема, што значи дека ентропијата е најмала. Информациска добивка е математички начин да се пресмета количината на информација која ја добиваме со избор на одреден атрибут од податочното множество и е претставена со равенката:

$$\textit{Information Gain(S,A)=Entropy(S)-}\sum_v \frac{|S_v|}{|S|} entropy(S_v)$$

S – Колкеција од тренинг примероците,

A – избраниот атрибут- еден од атрибутите во податочниот сет,

|*Sv*| - број на елементи во *Sv,*

|*S*| - број на елементи во *S,*

v - можните вредности на атрибутот,

$$\textit{Entropy= -}\sum_v p(v)\log_2 p(v),$$

каде *v* се можните вредности на соодветниот атрибут. Постојат повеќе алогритми за имплементација на дрво на одлучување како што се ID3, C4.5 или CART (Classification and regression trees). Овие алгоритми се разликуваат во метриката, односно параметрите кои ги користат за избор за атрибут од податочниот сет кој би им бил почетен, односно коренски јазол во дрвото на одлучување. Пример за овој алгоритам е даден на Слика 2.

Слика 2 Дрво на одлучување

Figure 2 Decision tree

4.4.1.1.2 К најблиски соседи (K nearest neighbors)

KNN (K - Nearest Neighbor) е техника за надгледувано учење со која се врши класификација на одредена податочна точка во дадена категорија, а сето тоа со помош на тренинг податочен сет. Овој алгоритам ги зема информациите од сите примероци во тренинг податоците и врз основа на тоа врши класификација на новите примероци притоа водејќи сметка за сличноста меѓу анализираниот примерок и најблиските примероци до него. Така, предвидувањата за секоја нова инстанца (x) се прават преку анализа и пребарување на целокупниот податочен сет за К најслични примероци (најблиски соседи) и сумирајќи ги резултантните променливи за избраните К инстанци.

Псевдо код за К најблиски соседи:

1. Вчитување на податочниот сет;
2. Иницијализација на вредноста на параметарот К;
3. Итерација од 1 до вкупниот број на тренинг податочни точки:
   - Пресметика на растојанието меѓу тест податоците и секој ред од тренинг податоците,
   - Сортирање на пресметаните растојанија во растечки редослед, врз основа на вредностите за оддалеченост,
   - Земање на првите К редови од сортираната низа,
   - Избирање на најзастапената класа од избраните редови во претходниот чекор,
   - Враќање на добиениот конечен резултат за предвидената класа.

4.4.1.2 Регресија (Regression)

Постојат два вида на регресија:

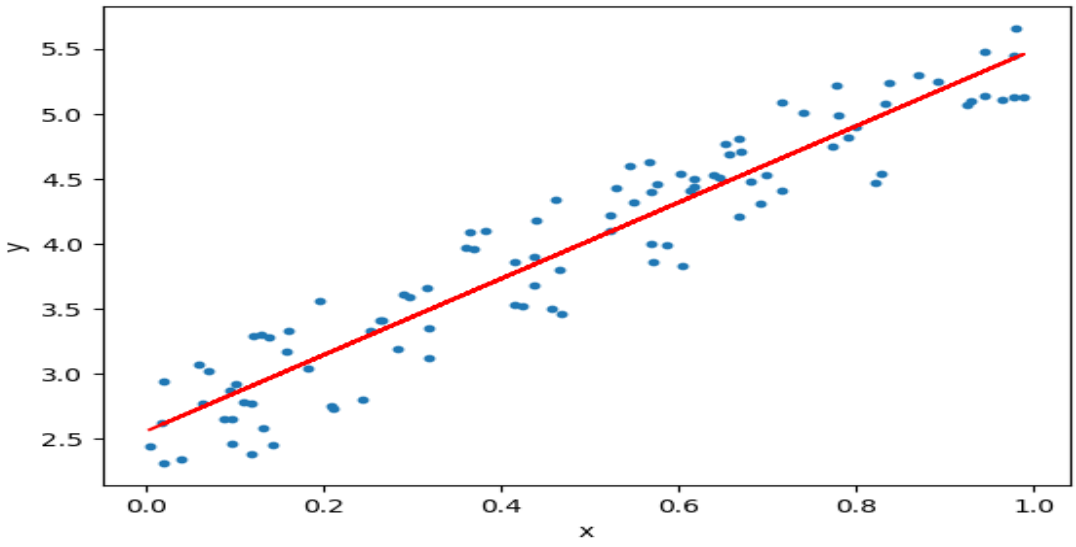
- линеарна регресија (Linear regression) и
- логичка регресија (Logistic regression).

4.4.1.2.1 Линеарна регресија (Linear regression)

Линеарната регресија е техника за машинско учење, која врши идентификација на линеарната врска меѓу променливите за објаснување од една страна и таргет променливите од друга страна и најчесто се користи за предвидување и прогнозирање на вредности притоа базирајќи се на историски податоци. Променливите чии вредности треба да се предвидат се наречени таргет промеливи, а променливите чии вредности помагаат, односно овозможуваат да се добие одредено предвидување - проценка за таргет променливите, се викаат објаснувачки променливи. Со линеарната врска, може да се идентификува влијанието на промените во објаснувачките променливи врз вредностите на таргет променливите. Дадена е равенката на линеарна регресија како: $y=\alpha x+\beta$

каде што: $\alpha=(N\sum xy-(\sum x)(\sum y))/(N\sum x^2-(\sum x)^2)$ $\beta=(\sum y-b(\sum x))/N$

Овде x и *y* се променливи кои го сочинуваат податочниот сет, а N е вкупниот број на вредности за променливите.



Слика 3 Пример за линеарна регресија

Figure 3 Example of linear regression

4.4.1.2.2 Логичка регресија (Logistic regression)

Логистичка регресија е еден од наједноставните и најчесто користените алгоритми за Машинско учење за класификација од две класи. Лесно е да се имплементира и може да се користи како основа за секој бинарен проблем на класификација. Логистичката регресија го опишува и проценува односот помеѓу една зависна бинарна променлива и независни променливи. Бинарната логичка регресија се соочува со ситуации во кои зависната променлива може да има една од две можни вредности. Меѓутоа, покрај бинарната, постои и друг вид на логичка регресија каде при решавањето на проблемот, резултантната променлива може да добие една од две или повеќе можни вредности.

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots\ldots + \beta nXn$$

y е линеарна функција од предиктивната променлива, *X* (X_1, X_2, *X3..*, X_n), која е слична на линеарната регресија. Значи резултатот од оваа функција ќе биде во рангот од 0 до 1. Врз основа на ова може да се одреди вредноста за бараната веројатност. Овде *y=logit(p),* а p е сигмоидната функција, исто така наречена логистичка функција, дава крива во облик на 'S' која може да има вредност помеѓу 0 и 1. Ако излезот од сигмоидната функција е повеќе од 0.5, можеме да го класифицираме исходот како 1 или ДА, а ако е помал од 0.5, можеме да го класифицираме како 0 или НЕ. На пример: Ако излезот е 0,75, може да се каже во смисла на веројатност како: „има шанса од 75 проценти дека пациентот страда од рак“.

$$p = 1/1 + e^{-y} \qquad p = 1/1 + e^{-(\beta 0 + \beta 1X1 + \beta 2X2 \ldots\ldots \beta nXn)}$$

4.4.2 Учење без надзор (Unsupervised learning)

Учењето без надзор е техника за учење на машини, каде што не треба да го надгледуваме моделот и треба да му дозволиме на моделот да работи самостојно за да открие информации. Ова учење главно се занимава со нерегистрирани податоци. Овој алгоритам за учење ви овозможува да извршувате посложени задачи за обработка во споредба со надгледуваното учење и учењето може да

биде понепредвидливо во споредба со другите методи на учење. Кај ваквата група на алгоритми за машинско учење, станува збор за покомплицирана метода бидејќи целта во овој случај е да се научи нешто без притоа да се даде некаков начин или патоказ за тоа како би се одвивал процесот на учење. Значи во случајов, не постојат таргет атибути кои би ги користеле за да вршиме споредба на нивните вредности со вредностите на примероците кои се предмет на анализа.
Еве кои се главните причини за користење на учењето без надзор:
- Неконтролирано машинско учење ги наоѓа сите видови непознати обрасци во податоците.
- Независните методи на ова учење ни помагаат да најдеме функции кои можат да бидат корисни за категоризација.
- Се случува во реално време, така што сите влезни податоци треба да се анализираат и обележат во присуство на ученици.
- Полесно е да се добијат необележани податоци од компјутер од етикетирани податоци, што бара рачна интервенција.
Постојат два пристапи на учење без надзор:

1. Кластерирање (Clustering) и
2. Асоцијација (Association).

4.4.2.1 Кластерирање (Clustering)

Кластерирањето е учење без надзор каде што основната функција е наоѓање на структура од колекција од неозначени податоци и е процес на организирање на објекти во групи чиишто членови се слични на одреден начин. Кластер претставува колекција од објекти кои меѓусебно се слични, а се разликуваат од објектите кои припаѓат на други кластери. Кластерирањето наоѓа примена во повеќе области: маркетинг каде имаме детерминирање на група од потрошувачи кои имаат слично однесување, врз основа на база на податоци која ги содржи податоците за нивните особини и навики на купување, проучување на земјотреси, односно кластерирање на набљудуваните епицентри заради идентификување на опасни зони, биологија како класификација на растенија или

животни базирајќи се на нивните особености. Најпознат и најкористен алгоритам за кластерирање е K-means алгоритмот.

4.4.2.1.1 K-means

Кај овој алгоритам најпрво се одредува бројот на кластери K и се дефинираат претпоставки за центрите на секој од кластерите кои се наречени центроиди. Било кој објект избран по случаен избор, може да биде иницијален центар. По одредување на центроидите, алгоритамот најпрво ги одредува централните координати, го одредува растојанието од секој објект до центарот и потоа врши групирање на објекти врз основа на минималното растојание. Процедурата што следи преставува избирање на едноставен начин за класифицирање на даден податочен сет во одреден број на кластери, дефиниран на почеток. Идејата е да се изберат К центроиди, по еден за секој кластер. Изборот на центроидите треба да се врши доста внимателно бидејќи од тоа зависи конечниот резултат. Токму поради ова, логично е да се изберат центроиди кои се сместени колку што е можно подалеку еден од друг. Следниот чекор е за секој од објектите кои припаѓаат во податочниот сет, да се пресмета растојанието до секој од центроидите, и врз основа на тоа да се направи поврзување меѓу содветниот објект и најблискиот до него центроид. Кога ова ќе се направи за секој од објектите во податочниот сет, тогаш е завршена првата фаза, односно е реализирано првичното групирање. Понатаму следува повторна пресметка на нови К центроиди, добиени како центари на новите кластери создадени во претходниот чекор. Откако ги имаме новите к центроиди, потребно е да се направи ново поврзување меѓу истите точки (објекти) од податочниот сет и најблискиот нов центроид. Така се генерира циклус во кој К центроиди ја менуваат својата локација чекор по чекор се додека постои објекти кои се движат, односно приближуваат до новосоздадените центроиди.

Освен тоа, алгоритмот е значајно чувствителен на иницијално и по случаен избор селектираните центри на кластерите. За да се намали овој ефект, K – means може да се изврши неколку пати, а потоа да се направи споредба на добиените резултати во секој од обидите и да се избере најдобриот. Дополнителна слабост

на овој алгоритам е тоа што резултатот зависи од вредноста на бројот на кластери – К. Сепак не постои генерално теоретско решение за наоѓање на оптималниот број на кластери за даден податочен сет. Иако постојат неколку методи кои се познати и користени за одредување на оптималниот број на кластери, сепак наједноставно решение во овој случај, е да се направат неколку обиди со различни вредности за бројот на кластери и да се изврши споредба на добиените кластери во секој од направените обиди.

4.4.2.2 Асоцијација (Association)

Уште еден вид на учење без надзор (Unsupervised learning) е асоцијацијата. Како што е кажано, кај овој тип на учење, целта не е примероците од податочниот сет да се класифицираат во една од предефинираните групи, туку во овој случај се бараат скриени релации и непознати врски во податочниот сет, сè со цел да се извлечат значајни информации. Целта на оваа техника е да се пронајдат одредени шеми, односно правила кои се појавуваат често, да се дефинираат врски или корелации меѓу множества од елементи во податоците.

За да го објасниме овој алгоритам на асоцијација ќе земеме еден пример на извршени трансакции на 5 купувачи во еден супермаркет. Множењето на вкупно купени производи во супермеркетот од страна на сите 5 купувачи се вкупно 5 , I = {млеко, леб, путер, пиво, пелени}, се прикажани во табелата како мала база на податоци што ги содржи производите, при што, во секој запис, вредноста 1 значи присуство на ставката (item) во соодветната трансакција, а вредноста 0 претставува отсуство на ставката во таа трансакција.

Број на трансакции	млеко	леб	путер	пиво	пелени
1	1	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	1
4	1	1	1	0	0
5	0	1	0	0	0

Табела 1 Пример за асоцијативно правило во супермаркет

Table 1 An example of an associate rule in a supermarket

Едно асоцијативно правило кое може да се изведе од Табела 1 е {путер,леб}⇒{млеко}, што значи дека ако се купи путер и леб, купувачите исто така купуваат млеко, каде што се претставени импликации во форма $X \Rightarrow Y$, каде X и Y се наречени ставки (itemsets), а $X \Rightarrow Y$ правило за асоцијација и T сет на трансакции од дадена база на податоци. Во продолжение ќе ги пресметаме од параметрите:

$Support(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$, ова е параметар кој покажува колку често се појавуваат ставките (items), во базата на податоци и во однос на T е дефиниран како дел од трансакциите t во базата на податоци што ја содржи ставката X. Во нашата база на податоци во примерот даден во Табела 1, ставката X={пиво,пелени} има вредност на $Support(X)=1/5=0.2$ бидејќи се појавува во 20% од сите трансакции (од 1 до 5 трансакција).

$(X \Rightarrow Y) = (X \cup Y) / Support(X)$, каде што овој параметар претставува мерка за тоа колку често одредена ставка се појавува во трансакција која истовремено содржи и друга ставка. Во нашиот пример, асоцијативното правило {путер,леб}⇒{млеко}, има мерка:

$Confidence$({путер,леб}⇒{млеко})=0.2/0.2=1, што значи дека 100% од трансакциите што содржат путер и леб, правилото е точно (100% од времето кога купувачот купува путер и леб, купува и млеко).

Забелешка: овој пример содржи мала база на податоци, а во практични апликации, ова правило има потреба од поддршка од неколку стотици трансакции, пред да може да се смета за статистички значајно, а базите на податоци често содржат илјадници или милиони трансакции.

5. Алатки за анализа и обработка на големи податоци

Големите податочни сетови, претходно се чувале во податочни магацини (Data Warehouse) во добро дефиниран формат- структурирани, често менаџирани со систем за управување со релациони бази на податоци (RDBMS). Наместо овие добро обработени и нормализирани податоци, огромната количина на податоци сега доаѓа како необработен, неструктуриран текст преку социјалните мрежи, паметните телефони, податоци од уредите за следење или пак уредите за идентификација на радио фрекфенциите. Најголемиот дел од податоците генерирани од новите технологии се добива во неструктуриран, сиров формат или во полуструктурирана форма што ги прави покомплексни за обработка и анализа. [18]. Денес се познати две софтверски решенија за анализа и обработка на големи податоци: Hadoop и NoSQL.

- Hadoop-служи за складирање и обработка на голами податоци во скалабилен и дистрибуиран модел на обработка.
- NoSQL-служи за обработка на бази на податоци, кој е оптимизиран за процесирање на огромни, неструктурирани и полуструктурирани податочни сетови.

5.1 Hadoop

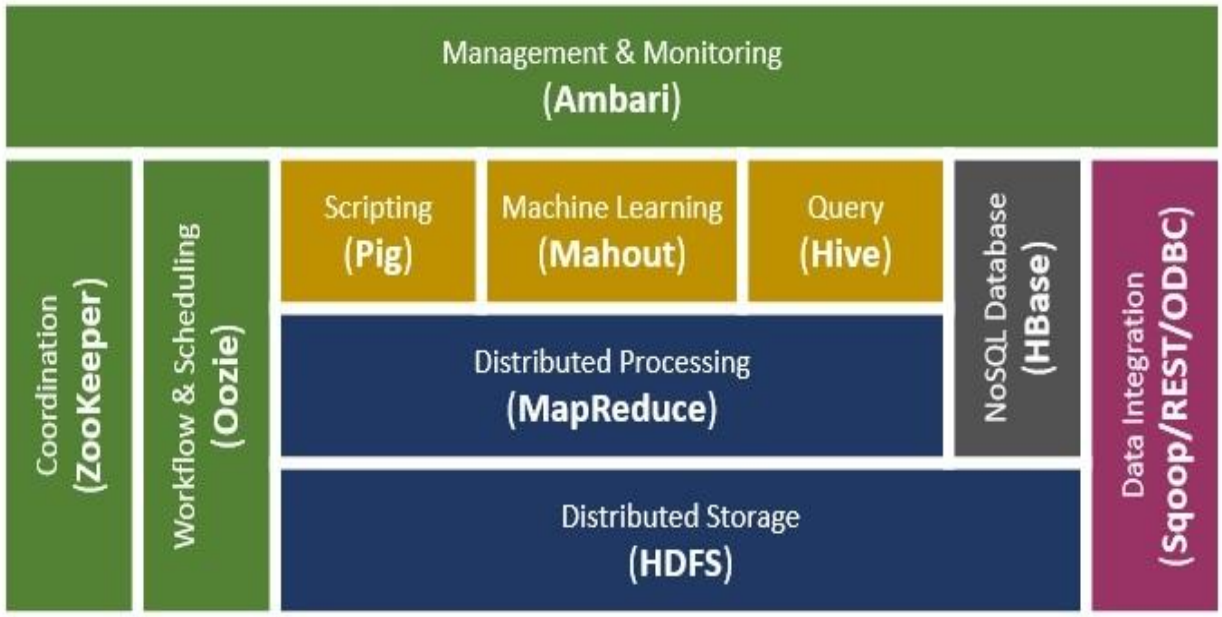
Apache Hadoop е позната технологија за големи податоци која е дизајнирана да ги избегне ниските перформанси и сложеноста со кои се соочуваме при процесирањето и анализата на големи податоци, користејќи традиционални технологии. Една главна предност на Hadoop е нејзиниот капацитет за брза обработка на големи множества на податоци, благодарение на неговите паралелни кластери и дистрибуираниот систем на датотеки. Всушност, за разлика од традиционалните технологии, Hadoop не ги копира сите далечни податоци за да ги изврши пресметките во меморијата. Наместо тоа, Hadoop ги извршува задачите каде се чуваат податоците. Така, Hadoop ја олеснува мрежата и серверите од значителна комуникациска оптовареност [19]. На пример, на

Hadoop потребни се само неколку секунди за да побараат податоци од неколку терабајти наместо 20 минути или повеќе на класичен систем. Друга предност на Hadoop е нејзината способност да извршува програми, истовремено обезбедувајќи толеранција на вина, обично се среќава во дистрибуираната околина. За да се гарантира тоа, го спречува губењето на податоци преку реплицирање на податоци на сервери.

Hadoop платформата се базира на две главни поткомпоненти:

- Hadoop дистрибуиран систем на датотеки (HDFS) и
- MapReduce.

Покрај тоа, корисниците можат да додадат модули на врвот на Hadoop колку што е потребно според нивните цели, како и нивните барања за примена (на пример, капацитет, перформанси, сигурност, приспособливост). Всушност, Hadoop придонесе да ја збогати својата околина со неколку модули со отворен код. На Слика 4 се дадени компонентите на Apache Hadoop:



Слика 4 Компоненти на Apache Hadoop

Figure 4 Apache Hadoop Components

5.1.1 Ниво на складирање на податоци

Во нивото за складирање на податоци на Hadoop спаѓаат:

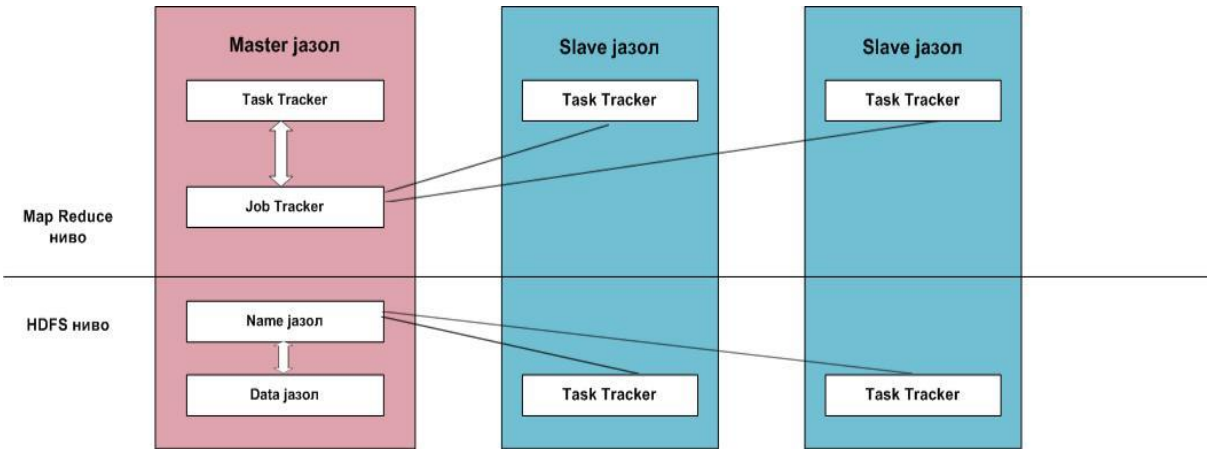
- Hadoop дистрибуиран систем на датотеки (HDFS) и
- Hbase.

5.1.1.1 Hadoop дистрибуиран систем на датотеки (HDFS)

HDFS е систем за чување податоци. Поддржува до стотици јазли во кластер и обезбедува економична и сигурна можност за складирање. Може да се справи со структурирани и неструктурирани податоци и да складира огромни количини (односно, складираните податоци може да бидат поголеми од терабајти). Сепак, корисниците мора да бидат свесни дека HDFS не претставува систем за општа намена. Ова е затоа што HDFS е дизајниран за обработка на сериите на датотеки со висока латентност. Покрај тоа, не обезбедува брзо пребарување на записи во датотеките. Главната предност на HDFS е нејзината преносливост во хетерогени хардверски и софтверски различни платформи. Покрај тоа, HDFS помага да се намали застојот на мрежата и да се зголемат перформансите на системот со поместување на пресметките во близина на складираните податоци. Таа обезбедува и репликација на податоци за поголема толеранција на грешка. Заради овие карактеристики е широко прифатен од многу корисници.

HDFS се заснова на архитектурата [20] како master/slave архитектура во која мастер јазолот се нарекува Name Node, додека пак slave делот од ваквата архитектура се состои од јазли наречени Data Node (податочни јазли). Мастер јазолот всушност е сервер кој управува со хиерахиската структура и пристапот (отворање, затворање, преименување) до документите од страна на клиентите. Начинот на кој фунцкионира овој јазол се состои во делење на влезниот податок во блокови и определување во кој податочен јазол ќе биде сместен секој од креираните блокови од податоци. Податочните јазли во себе ги содржат деловите од поделениот податочен сет, и истите ги праќаат кога за тоа ќе има побарување. Освен тоа, податочните јазли, го извршуваат и креирањето и бришењето на блокови од податоци. Hadoop, го дели податочниот сет на блокови кои се

складирани и се чуваат на повеќе податочни јазли (Data Nodes). Притоа, поради тоа што HDFS е дизајниран да поддржува големи фајлови, предефинираната големина на еден блок е 64 MB, која може да се зголеми во зависност од потребите [21].



Слика 5 Архитектура на HDFS и MapReduce нивото

Figure 5 HDFS and MapReduce layer architecture

HDFS функционира на тој начин што врши поделба на големите датотеки во помали делови наречени блокови. Овие блокови податоци се сместени на податочните јазли, а задачата на мастер јазолот (Name Node) е да знае кој од креираните блокови формираат една целина. Овој јазол исто така, игра улога на „сообраќаец" кој врши управување со целокупниот пристап до датотеките, вклучувајќи читање, запишување, бришење и репликација на блоковите податоци на податочните јазли. Податочните јазли не се „паметни" и тие постојано поставуваат прашања како на пример дали е потребно да се изврши некоја задача, до мастер јазолот. На овој начин, со вакво континуирано однесување, придонесува и му дава дополнителни информации на мастер јазолот за тоа кои податочни јазли се зафатени, а кои слободни. Од друга страна пак, податочните јазли комуницираат меѓусебно, па така тие можат да соработуваат за време на нормални операции во датотечниот систем. Овој начин на функционирање е неопходен бидејќи како што е кажано претходно, голема е веројатноста блоковите од една датотека да бидат складирани на повеќе различни податочни јазли.

Поради критичноста на мастер јазолот, потребно е истиот да биде реплициран, со цел да се создаде заштита од испад. Заради подобри перформанси, податочните јазли ги користат локалните дискови за складирање на податоците. Степенот на репликација и бројот на податочни јазли се дефинираат заедно со креирањето на кластерот, но бидејќи HDFS е динамичен систем, тоа значи дека секој од параметрите може да биде променет со цел да се прилагоди на новонастанатата состојба, за време на работењето на кластерот.

5.1.1.2 HBase

HBase [22] е дистрибуирана нерелациона база на податоци со отворен код, кој е изграден на врвот на HDFS. Таа е дизајнирана за операции со ниска латентност и е базирана на колонски ориентиран модел на клуч / вредност. Таа има потенцијал да поддржи високи стапки на ажурирање на табела и да ги распоредува хоризонтално во дистрибуираните кластери. HBase обезбедува издржлив структуриран хостинг за многу големи маси во формат на табела на големи податоци .

Табелите логично ги складираат податоците во редови и колони [23] и тие можат да се справат со билиони редови и милиони колони. HBase овозможува многу групи од атрибути да бидат групирани во семејства на колони, така што сите елементи од семејството на колони се зачувани заедно. Овој пристап е различен од релациона база на податоци ориентирана кон редови, каде што сите колони од редовите се складирани заедно и ова е повеќе изводливо отколку релациона база на податоци. Наместо тоа, HBase има предност што им овозможува на корисниците да воведат надградби за подобро да се справат со барањата за промена на апликациите. Сепак, HBase има ограничување дека не поддржува структуриран јазик за пребарување, како SQL.

Табелите на HBase се нарекуваат HStore и секој Hstore има една или повеќе мапирани-датотеки складирани во HDFS. Секоја табела мора да има шема со примарен клуч кој се користи за пристап до табелата. Редот се идентификува со име на табела и клуч за почеток, додека колоните може да имаат неколку

верзии за истиот клуч за редот. Hbase обезбедува многу опции како што се: пребарувања во реално време, пребарување на природен јазик, постојан пристап до извори на големи податоци, линеарна и модуларна приспособливост, автоматско и доследно обележување на табели. Тој е вклучен во многу решенија за големи податоци и веб-страници управувани од податоци, како што е платформата за пораки на Facebook. Слично на HDFS, HBase) има мастер јазол кој управува со кластери и секундарни јазли кои чуваат делови од табелите и извршуваат операции на податоци.

5.1.2 Ниво на обработка на податоци

Во нивото за обработка на податоци на Hadoop спаѓаат:

- Map Reduce и
- YARN.

5.1.2.1 Map Reduce

MapReduce [24] е рамка составена од програмски модел и нејзина имплементација. Тој е еден од првите основни чекори за новата генерација на алатки за управување и анализа со големи податоци и овозможува паралелна обработка. Всушност, програмскиот модел MapReduce ите на податоците: функцијата Map и функцијата Reduce.Поточно, програмата MapReduce се потпира на следниве операцкористи две последователни функции кои ги обработуваат пресметкии:

1. Прво, функцијата Мапа ги дели влезните податоци (на пример, долги текст датотеки) во независни податочни партиции кои претставуваат парови од клуч и вредност.

2. Потоа, MapReduce рамката ги испраќа сите парови од клуч и вредност во Map делот што ги обработува секоја од нив поединечно, низ неколку паралелни задачи на мапата низ кластерот. Секоја партиципација на податоци е доделена на единствен компјутерски јазол. На излез од Map, дава еден или повеќе средни парови од клуч и вредност. Во оваа фаза, задача на рамката е да ги собере сите

средни парови од клуч и вредност, за да ги сортира и групира по клучни зборови. Значи резултатот е листа со многу клучеви и придружни вредности.

3. Потоа, функцијата Reduce се користи за обработка на средно излезните податоци. За секој уникатен клуч, функцијата Reduce ги прифаќа вредностите поврзани со клучот во согласност со предвидената програма (т.е. филтрирање, сумирање, сортирање, земајќи просек или максимално) и дава еден или повеќе парови со клуч и вредност.

4. Конечно, рамката MapReduce ги зачувува сите излезни парови со клуч и вредности во излезната датотека.

Во рамките на MapReduce парадигмата, NameNode јазолот работи за Job Tracker за да закаже различни задачи и да дистрибуира задачи преку секундарните јазли. За да се осигура веродостојноста на извршувањето, JobTracker го следи статусот на секундарните јазли има задача на TaskTracker да додели задачи. Пример TaskTracker ги извршува задачите специфицирани од JobTracker и го следи нивното извршување. Секој TaskTracker може да користи повеќе Java виртуелни машини за да изврши неколку мапи или да ги намали задачите паралелно. Обично, кластерот Hadoop е составен од клиент сервер, повеќе DataNodes и два типа NameNodes (примарни и секундарни). Улогата на клиентскиот сервер е брзо да се вчитаат податоците, а потоа да се поднесат задачи за MapReduce на јазолот NameNode. Примарната NameNode јазол е посветен на координирање и управување со складирањето и пресметките. Од друга страна, второто име NameNode се справува со репликацијата и достапноста на податоците. Уникатен физички сервер може да се справи со трите улоги (клиент, примар и секундар) и роб) во мали групи (помалку од 40 јазли). Меѓутоа, во средни и големи кластери, секоја улога треба да биде доделена на една машина за сервери.

5.1.2.2 YARN

YARN е повеќе генерички од MapReduce. Таа обезбедува подобра способност за скала, подобрен паралелизам и напредно управување со ресурси во споредба со MapReduce. Тој нуди функции на оперативниот систем за

аналитички апликации на големи податоци и Hadoop архитектурата е изменета за да се вклучи YARN, YARN работи на врвот на HDFS. Оваа позиција овозможува паралелно извршување на повеќе апликации, но овозможува сериска и интерактивна обработка во реално време. YARN е компатибилен со интерфејс програмирани апликации (API) на MapReduce. Всушност, корисниците можат само да ги прекомплицираат работните задачи на MapReduce, за да ги извршуваат на YARN.

За разлика од MapReduce, YARN [25] ја подобрува ефективноста со делење на двете главни функционалности на JobTracker во два одделни делови:

(1) ResourceManager (RM) ги распределува и управува со ресурсите низ кластерот.

(2) Примарна апликативна рамка (AM) со библиотека, е дизајниран да закажува задачи, да ги совпаѓа со TaskTrackers и да го следи нивниот напредок. AM преговара со RM и Node Manager, обезбедува сметководствени задачи, одржува бројачи, ги рестартира неуспешните или бавните задачи и управува со животниот циклус на сите апликации извршени во кластерот.

5.1.3 Ниво на пристап до податоци

Во нивото на пристап до податоци во Hadoop спаѓаат:

- Pig;
- Hive;
- JAQL;
- Sqoop;
- Mahout;
- Flume;
- Chukwa.

5.1.3.1 Pig

Apache Pig [26] е рамка со отворен код која генерира виш скриптен јазик наречен Pig-Latin и ја намалува комплексноста на MapReduce со поддршка на

паралелното извршување на MapReduce задачи и задачи на Hadoop. Преку својата интерактивна средина, Pig Like Hive, едноставно истражува и обработува паралелни масивни множества на податоци, користејќи HDFS (на пример, комплексни податоци за ETL, различна анализа на податоци). Pig исто така дозволува интеракција со надворешни програми како што се shell скрипти, бинарни датотеки и други програмски јазици и има свој модел на податоци наречен Map Data (мапата е збир на парови од клуч и вредност). Pig Latin има многу предности. Се базира на интуитивна синтакса за поддршка на лесен развој на MapReduce задачи и го намалува развојното време додека поддржува паралелизам, така што корисниците можат да се потпрат на него и неколку оператори за да испраќаат и обработуваат податоци. Pig Latin е алтернатива на Java програмскиот јазик со скрипти слични на директен ацикличен график (DAG) и операторите кои обработуваат податоци претставуваат јазли, додека податоците се презентирани со рабови [27]. Спротивно, на SQL, Pig не бара шема и може да ги обработува полуструктурираните и неструктурираните податоци и поддржува повеќе формати на податоци од Hive.

5.1.3.2 Hive

Apache Hive [28] е систем за складирање на податоци дизајниран да ја поедностави употребата на Apache Hadoop. За разлика од MapReduce, кој управува со податоците во рамките на датотеките преку HDFS, Hive овозможува да ги претставува податоците во структурирана база на податоци што е повеќе позната за корисниците. Всушност, моделот на податоци Hive главно се базира на табели. Таквите табели претставуваат HDFS директориуми и се поделени на партиции. Секоја партиција потоа се дели на пласти. Покрај тоа, Hive обезбедува SQL-како јазик наречен HiveQL [29] кој им овозможува на корисниците пристап и манипулација со Hadoop-базирани податоци зачувани во HDFS или HBase. Hive не е погодна за трансакции во реално време бидејќи се базира на операциите со ниска латентност. Како и Hadoop, Hive е дизајниран за обработка на големи размери кои може да потраат неколку минути. Всушност, HiveQL транспарентно ги конвертира упитите (на пример: Ad hoc пребарувања, приклучува и сумирање) во

MapReduce задачи кои се обработуваат како групни задачи и овозможува приклучување на традиционалните MapReduce, кога не е изводливо или не е неопходно да се изразат во HiveQL. За разлика од повеќето SQL каде што имаат шема на запишување, Hive има шема на читање и поддржува повеќекратни шеми, што ја одложува примената на шема додека не се обидете да ги прочитате податоците. Предност е тоа што се вчитува побрзо, недостатокот е дека прашањата се релативно побавни и овде нема целосна SQL поддршка и не обезбедува вметнувања, ажурирања или бришење на ниво на ред.

5.1.3.3 JAQL

JAQL [30] е декларативен јазик на врвот на Hadoop кој обезбедува јазик за пребарување и поддржува масивна обработка на податоци ги конвертира барањата на високо ниво во JobReduce задачите. Тој беше дизајниран да ги пребаруваат полуструктурираните податоци врз основа на JSON (Java-Script Object Notation) формат, може да се користи за пребарување на други формати на податоци, како и многу типови на податоци (на пример, XML податоци одвоени со запирка (CSV) податоци). Значи, JAQL не бара шема на податоци како што е Pig. JAQL обезбедува неколку вградени функции, основни оператори и влезно-излезни адаптери. Ваквите карактеристики обезбедуваат обработка, складирање, преведување и конвертирање на податоци во JSON формат.

5.1.3.4 Sqoop

Apache Sqoop (Vohra, 2016) е софтвер со отворен код кој нуди командна интерфејс линија (CLI) кој обезбедува ефикасен пренос на гломазни податоци на линија за рефус помеѓу Apache Hadoop и структурирани податоци (како што се релациони бази на податоци и бази на податоци NoSQL). Sqoop нуди многу предности и обезбедува брзи перформанси, толеранција на грешки и оптимално искористување на системот за да ги намали оптоварувањата за обработка на надворешни системи. Трансформацијата на увезените податоци се врши со помош на MapReduce или било кој друг јазик на високо ниво како што се Pig, Hive

или JAQL (Jain, 2013). Овозможува лесна интеграција со HBase, Hive и Oozie. Кога Sqoop увезува податоци од HDFS, излезот ќе биде во повеќе фајлови. Овие команди може да бидат ограничени текст фајлови, бинарни или низа од фајлови кои содржат сериски податоци. Процесот на Sqoop Export ќе ги чита паралелно подесениот текст од HDFS паралелно, ги разгледува во записи и ќе ги вметнува како нови редови во табелата со целни бази на податоци.

5.1.3.5 Mahout

Apache Mahout [31] е моќна, скалабилна библиотека за автоматско учење која работи на врвот на Hadoop MapReduce. Машинско учење е дисциплина на вештачка интелигенција која им овозможува на системите да учат само врз основа на податоци, постојано подобрување на перформансите бидејќи се обработуваат повеќе податоци. Машинско учење е основа за многу технологии кои се дел од нашиот секојдневен живот. Некои примери на применети алгоритми за машинско учење вклучуваат:

Препораки: Бројни веб-сајтови денес се во можност да даваат препораки за корисниците врз основа на минатото однесување, како и однесувањето на другите. Netflix, на пример, може да препорача филм на корисник врз основа на неговата сличност со други филмови што ги гледал корисникот.

Филтрирање на спам: Речиси секој модерен провајдер на е-пошта може автоматски да ја открие разликата помеѓу спам-пораката и легитимната, само да ги претстави последните на корисникот. Овие филтрирања на мотори користат алгоритми за машинско учење, како кластерирање и класификација.

Обработка на природниот јазик: Многумина од нас имаат паметни телефони кои разбираат што мислиме кога бараме „Кога се следните играчи?“. Вклучувањето на компјутерот да ја разбере оваа фраза не е едноставна задача - мора да знае дека "niners" е сленг за San Francisco 49ers, кој е американски фудбалски тим, така што треба да се консултира со распоредот на Националната фудбалска лига за да го даде одговорот. Сето ова беше овозможено со примена на алгоритми за машинско учење за огромни множества на јазични податоци за да се направат

овие врски. До неодамна, научниците за податоци мораа рачно да ги имплементираат и прилагодат алгоритмите за машинско учење рачно до компјутерската рамка што ја користеа, што резултираше со значителна работа. Сега, со Hadoop и Mahout, научниците за податоци можат да напишат задачи на MapReduce кои се однесуваат на бројни предефинирани алгоритми за лесно изградување на овие видови апликации.

Скала: Покрај Java, Mahout корисниците ќе можат да пишуваат задачи, користејќи го програмскиот јазик Scala. Scala ги прави програмите за интензивна математика многу полесни во споредба со Јава, па програмерите ќе бидат многу поефикасни.

5.1.3.6 Flume

Flume (Хофман, 2015) е дизајниран да собира, складира и пренесува податоци од надворешни машини до HDFS. Таа има едноставна флексибилна архитектура и се справува со стриминг на податоци. Flume се базира на едноставен модел за да се справи со масовните извори на податоци кои не се изложени на податоци. Flume обезбедува различни карактеристики, вклучувајќи толеранција на грешка, механизам за сигурност кој може да се прилагодува, како и услуга за враќање на неисправности. Иако, Flume комплетно го надополнува Hadoop, тоа е независна компонента која може да работи и на други платформи. Познат е по својот капацитет да работи различни процеси на една машина. Користејќи Flume, корисниците можат да проследат податоци од различни извори и извори со голема јачина на звук (како Avro RPC извор и syslog) во тон (како што се HDFS и HBase) за анализа во реално време (Hoffman, 2013). Покрај тоа, Flume обезбедува машина за обработка на барањето која може да ја трансформира секоја нова податочна серија.

5.1.3.7 Chukwa

Chukwa (Ширеша и Бутада, 2016) е систем за собирање на податоци изграден на врвот на Хадооп. Целта на Chukwa е да ги следи големите дистрибуирани системи и користи HDFS за собирање на податоци од различни

даватели на податоци, а MapReduce ги анализира собраните податоци. Таа ја наследува приспособливоста и робусноста на Hadoop. Обезбедува интерфејс за отстранување, следење и анализа на резултатите. Chukwa нуди изводлива и моќна платформа за големи податоци. Таа им овозможува на аналитичарите да собираат и анализираат сетови на големи податоци, како и да ги следат и прикажуваат резултатите. За да се обезбеди веродостојноста, Chukwa е структуриран како нафтовод за собирање, фази на процесирање, како и депонирани интерфејси помеѓу фазите.

5.1.4 Ниво на управување со податоци

Во нивото на управување со податоци на Hadoop спаѓаат:

- Oozie;
- Ambari;
- Whirr;
- BigTop;
- Hue;
- ZooKeeper;
- Avro.

5.1.4.1 Oozie

Apache Oozie [32] е систем за обработка на работни процеси дизајниран да работи и да управува со работни задачи во кластери Hadoop. Тој е сигурен, проширен и скалабилен систем за управување кој може да се справи со ефикасно извршување на голем број на работни задачи. Работните задачи се во форма на директен ацикличен графици (DAGs), а може да поддржува различни типови на работа на Hadoop, вклучувајќи ги MapRe-duce, Pig, Hive и Sqoop работни. Една од главните компоненти на Oozie е серверот Oozie, кој се базира на две главни компоненти: Workflow Engine кој ги меморира и извршува различни типови работни задачи, како и Coordinator Engine кој работи со повторливи работни задачи предизвикани од предвидениот распоред. Oozie овозможува следење на

извршувањето на работните задачи, за корисниците да можат да го следат Oozie, со цел да го известат клиентот за работната состојба и статусот на извршување преку Http callbacks (на пример, работната задача е завршена, таа влегува или излегува од акциониот јазол). Во моментов, Oozie стандардно поддржува Derby како и други бази на податоци како: HSQL, MySQL, Oracle и PostgreSQL.

5.1.4.2 Ambari

Apache [33] е дизајниран да го поедностави управувањето со Hadoop благодарение на интуитивниот интерфејс, кој поддржува обезбедување, управување и следење на кластери на Apache Hadoop преку лесен веб-кориснички интерфејс за управување. Интерфејсот е базиран на RESTful API, како и други Hadoop компоненти како: HDFS, MapReduce, Hive, HCK, HBase, ZooKeeper, Oozie, Pig и Sqoop. Покрај тоа, Ambari обезбедува сигурност, користејќи протокол за проверка на автентичност, а исто така обезбедува функции за проверка на автентичност, овластување и ревизија базирани на улоги за управување со интегрираниот LDAP и Active Directory.

5.1.4.3 Whirr

Apache Whirr [34] го поедноставува создавањето и распоредувањето на кластери во облак средини, како што се Amazon AWS и обезбедува збирка на библиотеки за управување со овие услуги. Операторот може да работи со Whirr како алатка за командната линија, локално или во рамките на облакот. Покрај тоа, Apache Whirr поддржува обезбедување на Hadoop, како и Cassandra, ZooKeeper, HBase, Valdemort (складирање на клуч и вредности) и Hama кластери во облак средини.

5.1.4.4 BigTop

BigTop го поддржува Hadoop и има за цел да ги развие и да ги потврди субпроектите од Hadoop, како што се оние развиени од Apache. Целта е да се оцени и да се обезбеди интегритет и сигурност на системот како целина, наместо да се евалуира секој подмодул.

5.1.4.5 Hue

Hue [35] е веб-апликација за интеракција со Hadoop и ги содржи најчестите компоненти на Hadoop во еден интерфејс. Неговата главна цел е да им овозможи на програмерите да го користат Hadoop без да се грижат за командната линија. Hue помага да пребаруваме во системот, да создаваме и да управуваме со има за цел да помогне да се вчитаат, стартуваат и управуваат со податоци и да даваат резултати во Excel формат. Hue е компатибилен со било која верзија на Hadoop и е достапна во сите главни Hadoop дистрибуции.

5.1.4.6 ZooKeeper

Zookeeper е еден од субпроектите на Hadoop, чија главна задача е менаџирање на Hadoop, Hive, Pig, HBase, Solr и други проекти. Значи, тој игра улога на координациски сервис за дистрибуирани апликации. Доколку го анализираме од аспект на програмска логика, тогаш би можеле да кажеме дека е дизајниран од многу едноставен модел, многу сличен на структурата на систем за управување со директориуми. Се состои од два дела – клиент и сервер. Во еден кластер од сервери, само еден сервер ја има улогата на главен сервер, сите останати само чуваат копија од главниот, и во случај тој да не функционира, тогаш некој од нив продолжува со преземање на барањата од клиентите. Zookeeper клиентите се поврзани на серверот од сервисот на Zookeeper, а комуникацијата со серверот се остварува со праќање на барање и примање одговор од серверот,

преку TCP конекција. Со еден збор, Zookeeper претставува централизиран сервис за оддржување на конфигурациските информации и обезбедува дистрибуирана синхронизација и групни сервиси.

5.1.4.7 Avro

Apache Avro [36] и дефинира компактен и брз бинарен формат на податоци за поддршка на податоци и обезбедува поддршка за овој формат во различни програмски јазици, како што се Java, Scala, C, C ++ и Python. Avro обезбедува ефикасна компресија на податоци и складишта на различни јазли на Apache Hadoop. Во рамките на Hadoop, Avro пренесува податоци од една програма или јазик во друг (на пример, од C до Pig). Бидејќи податоците се складирани во својата шема, Avro е компатибилен со скриптните јазици. Avro шемите може да содржи и едноставни и сложени типови, а користи JSON како експлицитна шема или динамички генерира шеми на постоечките Java објекти.

5.2 Дистрибуции на Hadoop

Различни ИТ компании работат на подобрување и збогатување на инфраструктурата, алатките и услугите на Hadoop. Споделувањето на иновации на големи податоци преку модули со отворен код е корисно и промовира технологии за големи податоци. Сепак, недостатокот е дека корисниците можат да користат Hadoop платформа составена од различни верзии на модули од различни извори. Бидејќи секој Hadoop модул има различни карактеристики, постои ризик од несоодветност на верзиите во платформата Hadoop, при интеграцијата на разновидните технологии на иста платформа ги зголемува безбедносните ризици. Сепак, поголемиот дел од времето на комбинација на технологии од различни извори, може да донесе скриени ризици кои не се целосно истражени ниту тестирани. За да се соочат со овие проблеми, многу ИТ-продавачи како Cloudera, MapR и Hortonworks развија свои сопствени модули и ги спакуваа во сопствени дистрибуции. Една од целите е да обезбеди компатибилност, безбедност и перформанси на сите комбинирани модули.

Повеќето од достапните Hadoop дистрибуции постепено се збогатуваат и вклучуваат различни услуги како што се: дистрибуирани системи за складирање, управување со ресурси, услуги за координација, интерактивни алатки за пребарување, напредни алатки за анализа на разузнавачки информации итн.

5.2.1 Cloudera

Cloudera [37] е една од најчесто користените Hadoop дистрибуции. Таа има многу предности, како што се централизирана административна алатка, унифицирана обработка на серии, интерактивен SQL, како и контрола на пристап базирана на улоги.Освен тоа, решенијата на Cloudera можат да се интегрираат во широк опсег на постоечка инфраструктура и можат да се справат со различни оптоварувања и формати на податоци во еден систем. Cloudera нуди лесен начин за прегледување и пребарување на податоци во Hadoop. Всушност, можно е да се реализира интерактивно пребарување во реално време и да се визуелизираат резултатите на пригоден начин.

Еден од Cloudera модулите е Impala. Тој претставува интересен јазик за пребарување кој е компатибилен со Hadoop и структурира податоци во формат во вид на колони на податоци. Тоа овозможува да се справи со интерактивни големи податоци во реално време. Спротивно на Hive, Импала не ја користи рамката MapReduce, туку тој користи свој процесор за обработка и меморирање, за да обезбеди брзи пребарувања преку големи множества на податоци. Така, Импала може директно да ги користи податоци од постоечките HDFS и HBase извори и го минимизира движењето на податоците и со тоа го намалува времето на извршување на поставените задачи. Cloudera обезбедува и флексибилен модел кој поддржува структурирани, како и неструктурирани податоци. Cloudera е побрз од Hive и извршува барања најмалку 10 пати побрзо од Hive / MapReduce. Констатирано е дека во споредба со HiveQL (Hive Query Language), Cloudera обезбедува 7 до 45 пати поголеми перформанси за барања за агрегација на податоци кои беа забрзани дури за околу 20 до 90 пати. Cloudera, исто така, ја надминува HiveQL или MapReduce во однос на реактивноста во реално време:

всушност, верзијата на Cloudera Enterprise го намалува времето на одговор на барањата во секунди, наместо во минути како кај HiveQL или MapReduce. И покрај сите наведени предности, Cloudera има некои недостатоци. На пример, не е погодно за пребарување на стриминг податоци, како што се стриминг видео или континуирани податоци за сензори.

5.2.2 MapR

MapR [38] е комерцијална дистрибуција за Hadoop дизајнирана за претпријатија. Подобрена е за да обезбеди подобра сигурност, перформанси и лесно складирање на големи податоци, обработка и особено анализа со алгоритми за машинско учење. Обезбедува збир на компоненти и модули кои можат да се интегрираат во широк спектар на Hadoop. MapR не користи HDFS, всушност, MapR има развиено сопствени MapR File Systems (MapR-FS) со цел да ги зголеми перформансите и да овозможи лесно враќање на податоците. На MapR-FS има предност да биде компатибилен со NFS. Така, податоците може лесно да се пренесат меѓу нив. MapR е базиран на стандардниот програмски модел на Hadoop.

5.2.3 Hortonworks

Hortonworks податочната платформа (HDP) [39] е изградена на Apache Hadoop за да се справи со складирање на големи податоци, пребарување и процесирање. Тој има предност во поглед на тоа што обезбедува брзо, економично и скалабилно решение и неколку услуги за управување, следење и интеграција на податоците. Покрај тоа, таа е позиционирана како клучна интегративна платформа, бидејќи обезбедува алатки за управување со отворен код и поддржува врски со некои BI платформи.

HDP обезбедува дистрибуирана меморија преку DHFS и нерелациона база на податоци Hbase. Овозможува дистрибуирана обработка на податоци врз основа на MapReduce, пребарувајќи податоци преку Hue и брзи скриптина податоци кои

користат Pig. HDP вклучува Oozie за управување и распоред на работни задачи, како и Hcatalog за обработка на услугите за големи податоци. Многу алатки се исто така достапни во HDP, вклучувајќи webHDFS, Sqoop, Talend Open Source, Ambari и Zookeeper.

6. Истражување

Во мојот магистерски труд анализирана беше базата на податоци на системот за е-учење на Moodle на Универзитетот „Гоце Делчев“ во Штип. Таа има MySQL база на податоци со повеќе од 300 табели кои содржат податоци од 7 години (од 2012 г. до 2019 г.) и нејзината големина е околу 13GB. Имаме големи податоци за подолг временски период. Тоа е добра основа за анализирање и извлекување на знаења за однесувањето на наставниците. Поради тоа што станува збор за прилично долг период, но и систем со голем број на корисници кои претставуваат извор на секојдневно генерирање на нови податоци, беше неопходно користење на соодветни алатки за процесирање на огромната база на податоци. Затоа најпрво ќе дадеме некои основни карактеристики на системот за електронско учење – Moodle како и неговата архитектура и основни елементи.

6.1 Moodle систем

Moodle (*Modular Object-Oriented Dynamic Learning Environment*) претставува софтверска платформа за електронско учење [40]. Moodle е систем за електронско учење, кој се користи од страна на десетици милиони луѓе ширум светот, како образовните институции за учење на далечина и се користи од страна на претпријатија низ целиот свет за online обука и учење. Според некои автори Moodle е систем за менаџирање со курсеви (CMS - Course management system), според други е систем за управување со учењето (LMS - learning management system) или пак виртуелна околина за учење (VLE -virtual learning environment). Од неодамна Moodle се дефинира како систем за управување со содржини за учење (LCMS– learning content management system). Ваквиот вид на

систем овозможува креирање, складирање, ажурирање и праќање на содржина за електронско учење која може да биде персонализирана. Овие содржини се пренесуваат во форма на таканаречени објекти за учење. LCMS ги комбинира карактеристиките на CMS и LMS. Moodle претставува платформа која овозможува пристап до содржини за учење, креирање курсеви, комуникација на форум, доставување на задачи и многу други активности. Moodle, односно неговата структура се базира на апликациско јадро и голем број на додатни компоненти (plugins), а како клучни елементи се курсеви, активности и корисници. Во овој систем за електронско учење, секој кој го користи системот, односно има пристап до него се нарекува корисник (user). За да биде запишан на одреден курс, на секој од корисниците треба да му биде доделена одредена улога – наставник или студент. Moodle платформата се состои од модули, кои претставуваат интеракциска врска меѓу професорите и студентите, како што се квизови, форум, задачи, анкети, лекции, речник и други. Moodle разликува три вида на интеракции и тоа: студент-студент, студент-наставник и студент-содржина. Базата на податоци во Moodle содржи повеќе од 250 табели.

6.2 Фази на истражувачката работа

1. Собирање на податоци;
2. Препроцесирање и трансформација на собраните податоци и креирање на податочен сет погоден за понатамошна обработка;
3. Примена на алгоритмите за податочно рударење (кластерирање - K means);
4. Евалуација и анализа на добиените резултати и
5. Донесување заклучоци.

6.2.1 Собирање на податоци

Во мојот магистерски труд анализирана беше базата на податоци на системот за е-учење на Moodle на Универзитетот „Гоце Делчев“ во Штип. Таа има

MySQL база на податоци со повеќе од 300 табели кои содржат податоци од 7 години (од 2012 г. до 2019 г.) и нејзината големина е околу 13 GB. Имаме големи податоци за подолг временски период. Јас треба да направам селекција на потребните податоци кои ќе бидат основа на целокупната понатамошна работа и истражувања, така што е неопходно да се познаваат особеностите на податоците кои се добиени од едукативните системи, информациите можат да доаѓаат од различни извори на податоци, може да има некомплетни податоци и загуба на податоци, корисниците во системот, односно наставниците јасно се идентификувани, постојат прилично голем број на достапни инстанци и атрибути кои може и да не се доволно корисни за анализата, па поради тоа е потребно да се направи избор на оние најпотребните и често е потребно да се направи дискретизација на нумеричките вредности - броевите, заради подобрување на разбирливоста на податоците и добиените модели.

Тоа е добра основа за анализирање и извлекување на знаења за однесувањето на наставниците и затоа фокусот на истражувањето е ставен на активностите на наставниците. Поради постоење на голем број на табели во базата на податоци во Moodle, ги избрав табелите кои можат да ни помогнат да ја откриеме активноста на наставниците на платформата. Најпрво беа избрани потенцијални табели за истражување од кои понатаму со соодветни упити и пребарувања ќе се извлечат потребните податоци со анализа на атрибутите кои ги поседуваат табелите во базата на податоци, како и релациите кои постојат меѓу нив. Табели што ги користев во моето истражување се:

-mdl_logstore_standard_log – сите интеракции на посетители се запишани во оваа табела, вклучувајќи ги активностите на наставникот;

-mdl_user – секој корисник на Moodle има запис во оваа табела (вклучувајќи ги информациите за наставниците);

-mdl_role_assignment – улогата на корисникот во Moodle;

-mdl_assign – информација за задачата која креаторот на курсот(наставникот) ја вклучил;

-mdl_assign_submission – записи за секое доставување на задача;

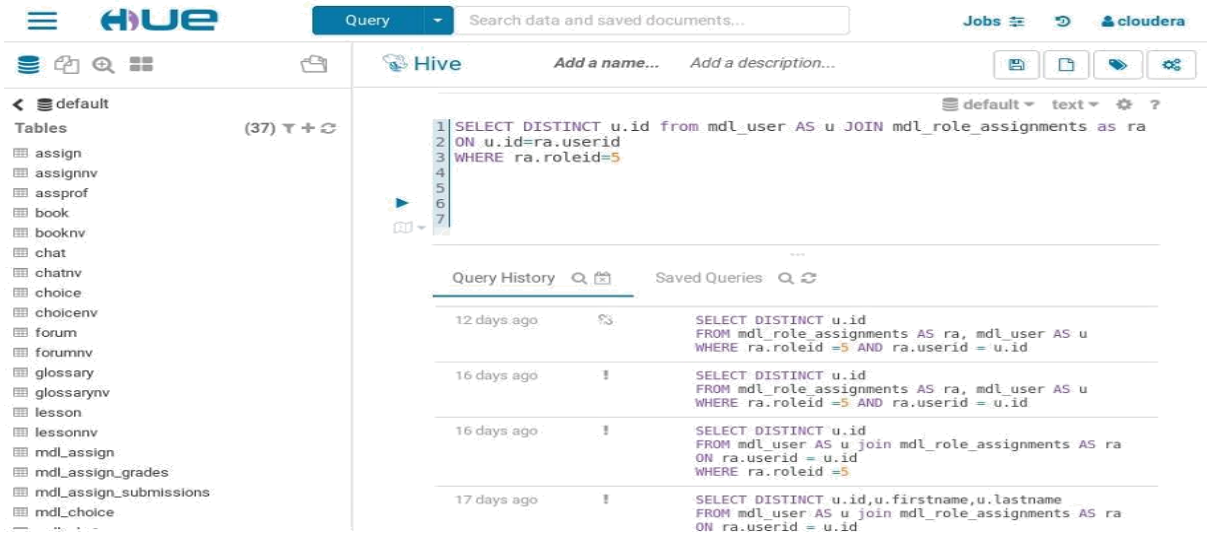
-mdl_assign_grades – информации за оценките за секоја од задачите;

-mdl_course – за секој нов креиран курс на Moodle, се додава запис во оваа табела;
-mdl_forum – информации за секој креиран форум;
-mdl_discussions – записи за дискусиите водени на форуми;
-mdl_posts – информации за секој коментар;
-mdl_quiz – информации за активностите во врска со квиз;
-mdl_choice – секоја инстанца на активност од видот избор, се запишува во оваа табела;
-mdl_lesson – за секоја активност во врска со одредена лекција, се прави нов запис во оваа табела.

Притоа, мора да се напомене дека во оваа фаза на собирање на податоци, е важен изборот на табели бидејќи погрешниот избор на табели или табели кои не содржат доволно информации го отежнува процесот на истражување и понатамошна работа.

6.2.2 Препроцесирање и трансформација на собраните податоци и креирање на податочен сет погоден за понатамошна обработка

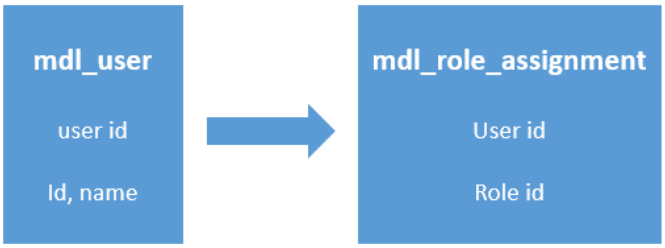
Честопати, податочниот сет добиен после фазата на собирање на податоци треба да се подготви за понатамочна анализа, треба да се изврши поврзување на потребните табели и задавање на соодветни упити, заради извлекување на потребните податоци. Jac ја користив платформата Hadoop и со помош на алатката Sqoop се озвоможува и креирање на табела во Hue делот од Hadoop, Слика 6, каде се прикажува како Hive табела. Вака импортирани, на табелите се извршуваат потребните HiveQL упити со цел извлекување на бараните податоци:



Слика 6 Hue-делот од Hadoop со импортираните табели од Moodle базата на податоци

Figure 6 Hadoop hue section with the imported tables from the Moodle database

Со цел да се олесни работата за добивање активност на наставниците (асистентите и професорите) во Moodle, создадов привремени (помошни) табели. Прво, јас создадов табела со сите наставници. За таа цел, ги користив табелите "mdl_user" и "mdl_role_assignments" кои се поврзани со атрибутот "userid" (Слика 7). Ги избрав само оние корисници кои имаат улога на наставник користејќи ја табелата "mdl_role_assignment". Со ова, добив нова табела со атрибути "userid", "firstname", "lastname" и "role". За испитуваниот период, од 2012-2019 година, добиен е резултат од 336 наставници чиишто активности ќе бидат предмет на обработка на ова истражување.



Слика 7 Добивање на табела на наставници

Figure 7 Getting a table with teachers

Користејќи ги табелите од базата на податоци, особено табелата со логирани кориснички активности (mdl_logstore_standard_log), создадов привремени табели кои ги содржат податоците за индивидуалните корисници за

одредена активност на некој модул. Извршено беше истражување на наставниците за разни модули на Moodle, како што се форуми, курсеви, речник, задачи, анкети, квизови, избори, чатови, вики и книги. По ова, се приклучивме на табелата која ги содржи наставниците и табелите со активностите на наставникот со користење на прашањата и атрибутот "userid" во сите табели (Слика 8).

Слика 8 Блок-дијаграм за добивање на сите активности на наставниците

Figure 8 *Workflow for obtaining all activities of the teachers*

Од сите добиени активности на наставниците, јас ќе ги прикажам селектираните атрибути кои беа испитувани за време на ова истражување:

assign, assign NV - број на завршени задачи и број на активности во модулот задачи кои се разликуваат од преглед на содржините во овој модул;
book, book NV - број на активности во модулот книга и број на активности кои се разликуваат од едноставен преглед на содржините во овој модул;
chat, chat NV – број на реализирани разговори во модулот за разговор;
choice, choice NV - број на активности во модулот избор и број на активности кои не вклучуваат преглед на содржините во овој модул;
forum, forum NV - број на активности во модулот форум и број на активности кои вклучуваат ативно учество во комуникацијата;
glossary, glossary NV - број на активности во модулот речник и број на активности кои се разликуваат од едноставен преглед на содржините во овој модул;
quiz, quiz NV - број на активности во модулот квиз и број на активности кои се разликуваат од пасивен преглед на содржините во овој модул;

survey, survey NV - број на активности во модулот анкета и број на активности кои се разликуваат од едноставен преглед на содржините во овој модул.

Со задавање на HiveQL упит во Hue делот од Hadoop платформата:

```
select a.userid, a.firstname,a.lastname, a.`role`,a.faculty,f.c1, fnv.c1,ass.c1,assnv.c1,
bk.c1,bknv.c1,ch.c1,chnv.c1,cho.`_c1`,chonv.c1,glo.c1,glonv.c1,less.c1,lessnv.c1,qu.c1
,qunv.c1,su.c1,sunv.c1,wik.c1,wiknv.c1
from assprof a left JOIN forum f on(a.userid=f.userid) left JOIN forumnv fnv
on(a.userid=fnv.userid)
    left JOIN assign ass on(a.userid=ass.userid) left JOIN assignnv assnv
    on(a.userid=assnv.userid)
    left JOIN book bk on(a.userid=bk.userid)
    left JOIN booknv bknv on(a.userid=bknv.userid)
    left JOIN chat ch on(a.userid=ch.userid)
    left JOIN chatnv chnv on(a.userid=chnv.userid)
    left JOIN choice cho on(a.userid=cho.userid)
    left JOIN choicenv chonv on(a.userid=chonv.userid)
    left JOIN glossary glo on(a.userid=glo.userid)
    left JOIN glossarynv glonv on(a.userid=glonv.userid)
    left JOIN lesson less on(a.userid=less.userid)
    left JOIN lessonnv lessnv on(a.userid=lessnv.userid)
    left JOIN quiz qu on(a.userid=qu.userid)
    left JOIN quiznv qunv on(a.userid=qunv.userid)
    left JOIN survey su on(a.userid=su.userid)
    left JOIN surveynv sunv on(a.userid=sunv.userid)
    left JOIN wiki wik on(a.userid=wik.userid)
    left JOIN wikinv wiknv on(a.userid=wiknv.userid)
group by a.userid, a.firstname,a.lastname,a.`role`,a.faculty,f.c1, fnv.c1,ass.c1,assnv.c1,
bk.c1,bknv.c1,ch.c1,chnv.c1,cho.`_c1`,chonv.c1,glo.c1,glonv.c1,less.c1,lessnv.c1,qu.c1
,qunv.c1,su.c1,sunv.c1,wik.c1,wiknv.c1 order by a.firstname desc;
```

Со ова, добив табела со која се врши спојување на секоја од помалите помошни (привремени) табели, или поттабели во кои се наоѓаат информации за вкупниот број на активности на секој од наставниците:

userid	f.c1	fnv.c1	ass.c1	assnv.c1	bk.c1	bknv.c1	ch.c1	chnv.c1	cho_c1
14778	3	NULL	69	1	NULL	NULL	NULL	NULL	NULL
896	974	440	519	132	NULL	NULL	NULL	NULL	NULL
28922	10	NULL	9	NULL	NULL	NULL	NULL	NULL	14
8906	11	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8893	534	257	NULL	NULL	NULL	NULL	NULL	NULL	5
10448	26	4	NULL	NULL	NULL	NULL	NULL	NULL	NULL
30279	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8803	193	15	191	NULL	NULL	NULL	NULL	NULL	NULL
8805	422	210	502	12	NULL	NULL	NULL	NULL	NULL
9058	3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9062	218	81	963	622	NULL	NULL	1	NULL	NULL
4235	906	377	NULL	NULL	NULL	NULL	NULL	NULL	NULL
9031	1159	523	NULL	NULL	NULL	NULL	NULL	NULL	NULL
30385	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4174	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
17341	57	24	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10466	37	14	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10463	11	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8990	179	56	1	NULL	NULL	NULL	NULL	NULL	NULL
8971	4	2	NULL	NULL	NULL	NULL	NULL	NULL	NULL
11073	249	3	168	46	NULL	NULL	NULL	NULL	3
8921	11	NULL	6	NULL	NULL	NULL	1	NULL	8

Табела 2 Вкупните активности на наставници

Table 2 Total teacher activities

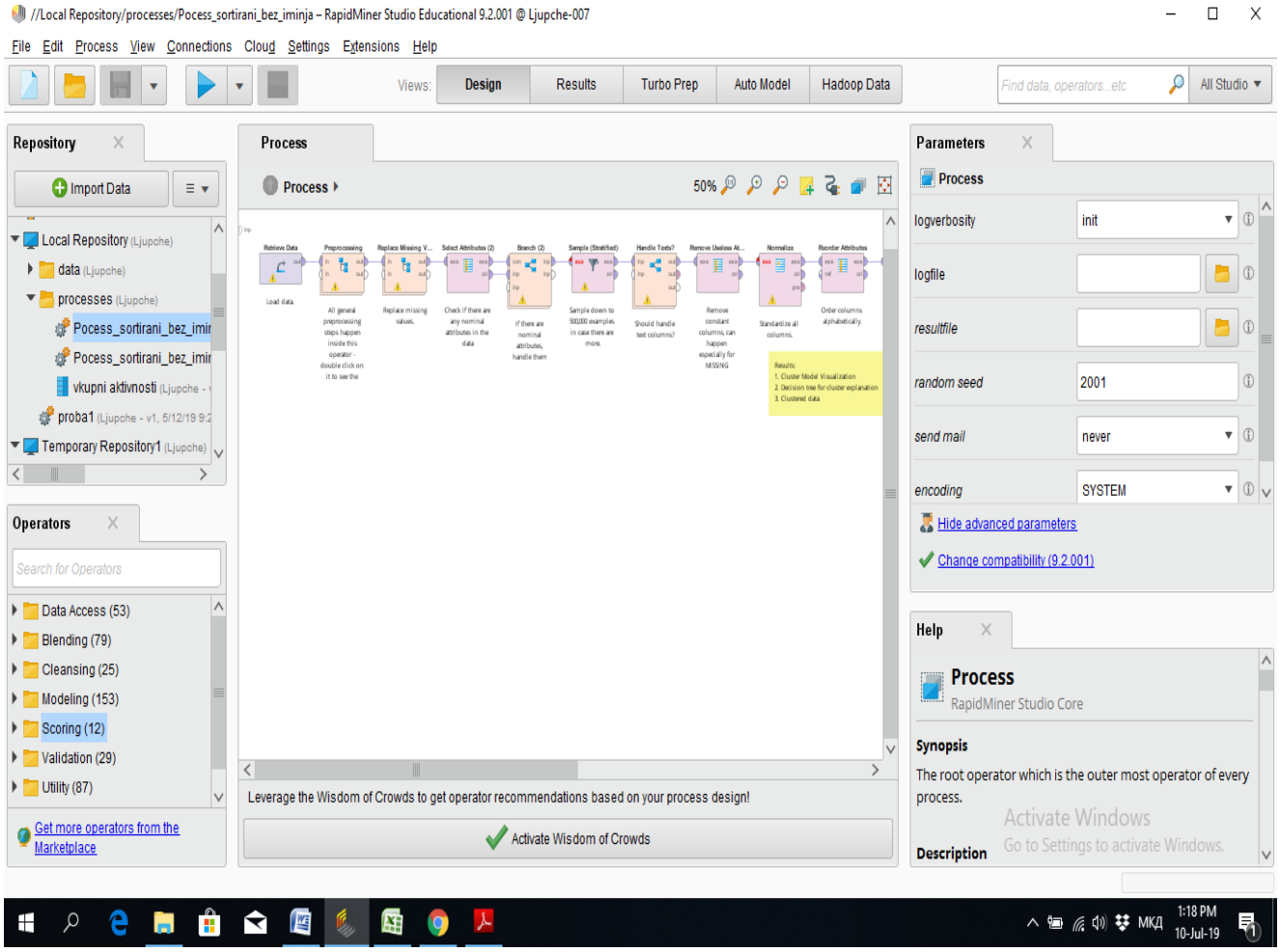
Така да оваа Табела 2 е всушност конечниот податочен сет (assprof.csv) во csv формат, кај која овде не се прикажани колоните (a.firstname, a.lastname,a.role, a.faculty), заради законот на приватност на наставниците, притоа, како активности за анализа се селектирани активностите во модулите - assign, book, chat, choice, forum, glossary, quiz и survey, кој така подготвен понатаму е користен во следната

фаза од истражувачката работа - примена на алгоритми за податочно рударење. Овој податочен сет во себе содржи голем број NULL вредности и тие беа соодветно заменети и прочистени на начин на кој не би влијаеле негативно врз крајниот резултат од анализата.

6.2.3 Примена на алгоритми за податочно рударење (Кластерирање со K-means)

Јас во ова истражување, со оглед на видот на податоците со кои располагам, одлучив да направам дескриптивната анализа, поточно описна анализа, која врши сумаризација на необработените податоци и од нив создава нешто што е читливо и разбирливо и од големата количина податоци се извлекуваат значајни информации и се врши кластерирање на податочниот сет, со цел да се утврди постоење на неколку групи на наставници кои имаат слични карактеристики во однос на користењето на Moodle, но и да се констатираат одредени правила во однесувањето на секоја од дефинираните кластери од наставници. Дополнително, може да се направи анализа на користењето на системот за електронско учење за време на испитуваниот период, односно за период за кој крајните испитувања не се завршени. Со групирањето на наставниците врз основа на нивните активности на системот за електронско учење, би можеле да се детерминираат наставници за кои постои можност од постигнување на недоволни резултати на финалните испитувања. На овој начин, се определува таргет група на наставници за кои може да се обидат да пронајдат начини за дополнителна мотивација за недоволно активните студенти, или пак да се обидат да им помогнат во надминување на проблемите во учењето на студентите. Во овој дел од истражувачката работа, беше направена имплементација на кластерирање со алгоритамот K-means. За таа цел во мојата истражувачка работа јас ја користив алатката RapidMiner Studio за предиктивна анализа за машинско учење и за визуелизација на добиените резултати. Најпрво

јас го импортирав податочниот сет assprof.csv во околината на RapidMiner Studio. На Слика 9 е прикажан како изгледа RapidMiner Studio.

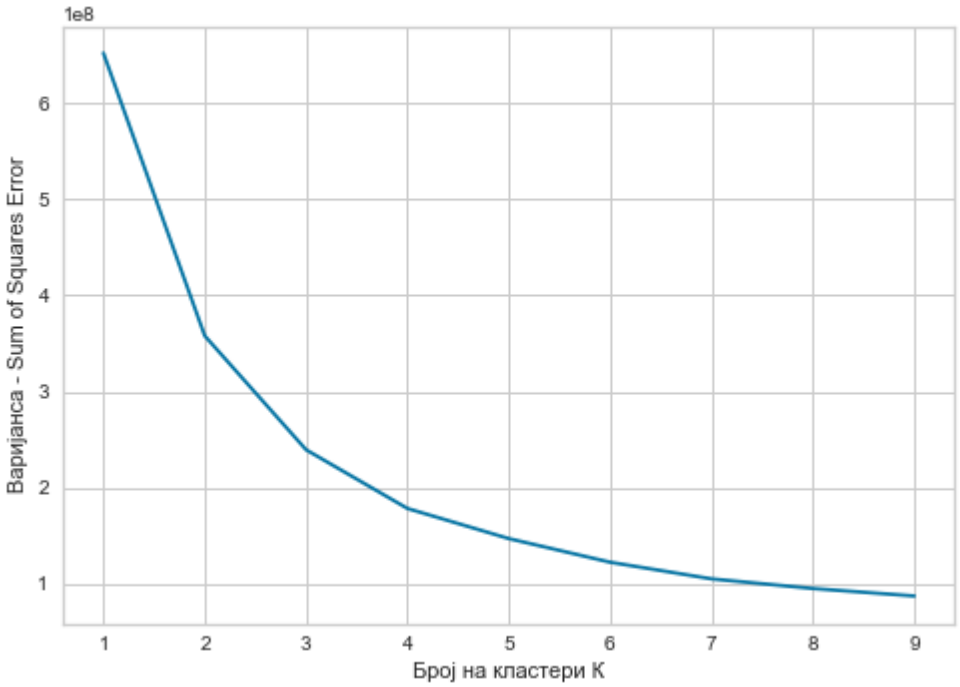


Слика 9 RapidMiner Studio

Figure 9 RapidMiner Studio

За кластерирањето како вид на машинско учење без надзор, еден од најчесто користените алгоритми за кластерирање е K-means. Значен елемент во примената на овој алгоритам, којшто може да влијае на крајните добиени резултати, е изборот на K - бројот на кластери. За одредување на оптимална вредност на K, постојат неколку методи, меѓу кои доста често е употребуван *Elbow* методот [41], кој беше користен за да се направи имплементација на K-means со најдобро избрана вредност за бројот на кластери. Овој метод функцинира на

начин што врши пресметка на вредноста на варијансата (сума од квадрати) внатре во секој од кластерите, а тоа го прави за различни вредности на бројот на кластери.



Слика 10 Примена на Elbow метод при избор на бројот на кластери K

Figure 10 Application of the Elbow Method when selecting the number of clusters K

Од графичкиот приказ, може да се утврди дека со зголемување на бројот на кластери доаѓа до намалување на варијансата, меѓутоа интензитетот на намалувањето на варијансата опаѓа откако вредноста на бројот на кластери е 3 или 4. Поради тоа што не можеме со сигурност да тврдиме дека добиените резултати за оптимален број на кластери се со сигурност точни, односно би ни дале оптимален и најдбоар резултат во кластерирањето, најдобро е да се направат неколку обиди за кластерирање со 3 или 4 кластери а потоа да се направи споредба на добиените резултати. Од направените обиди заклучив дека најдобар резултат се добива кога оптималната вредност на бројот на кластери е K=3 и јас ја земав оваа вредност во понатамошната анализа.

Но пред да се направи кластерирање на податочниот сет assprof.csv, најпрво ја генерирав матрицата на корелација на атрибутите, прикажана на Слика11. Корелацијата е статистичка мерка за релацијата меѓу две променливи. Значи, оваа матрица е приказ на поврзаноста меѓу атрибутите во податочното множество, а истата е презентирана како вредност од -1 до 1. Доколку прикажаната вредност во матрицата за два атрибута е 0, тогаш тоа значи дека не постои никаква поврзаност меѓу тие атрибути. Негативна корелација пак значи дека доколку вредноста на едниот од атрибутите се зголемува тогаш вредноста на вториот атрибут се намалува. Позитивна корелација пак, означува дека вредностите и на едниот и на другиот атрибут се движат во иста насока.

	assign	assign nv	book	book nv	chat	chat nv	choice	choice nv	forum	forumnv	glossary	glossary nv	lesson	lesson nv	quiz	quiz nv	survey	userid	wiki	wikinv
assign	1	0.874	-0.012	-0.008	0.456	0.355	0.435	0.281	0.065	0.057	0.2536	0.225	-0.013	-0.013	0.176	0.109	0.118	-0.07	-0.006	-0.006
assign nv	0.874	1	-0.01	-0.007	0.596	0.543	0.583	0.339	0.02	0.0099	0.1046	0.089	-0.008	-0.008	0.108	0.049	0.126	-0.043	-0.007	-0.007
book	-0.01	-0.01	1	0.987	-0.01	-0.01	-0.01	-0.01	-0.035	-0.033	-0.011	-0.01	-0.004	-0.004	-0.01	-0.01	0.009	-0.067	-0.004	-0.004
book nv	-0.01	-0.01	0.9871	1	-0.01	-0.01	-0.01	-0.01	-0.028	-0.026	-0.009	-0.01	-0.003	-0.003	-0	-0	0.002	-0.071	-0.003	-0.003
chat	0.456	0.596	-0.013	-0.01	1	0.804	0.66	0.406	0.137	0.1172	0.0376	0.076	-0.009	-0.009	0.235	0.245	0.153	-0.038	-0.009	-0.009
chat nv	0.355	0.543	-0.007	-0.005	0.804	1	0.647	0.29	-0.004	-0.009	-0.012	-0.01	-0.005	-0.005	0.046	0.002	0.083	-0.024	-0.005	-0.005
choice	0.435	0.583	-0.009	-0.008	0.66	0.647	1	0.707	0.127	0.1516	0.107	0.204	-0.007	-0.007	0.414	0.362	0.134	-0.019	-0.007	-0.007
choice nv	0.281	0.339	-0.006	-0.008	0.406	0.29	0.707	1	0.089	0.1139	0.1037	0.187	-0.007	-0.007	0.417	0.466	0.088	0.0557	-0.007	-0.007
forum	0.065	0.02	-0.035	-0.028	0.137	-0	0.127	0.089	1	0.9359	0.1634	0.146	-0.032	-0.032	0.174	0.157	0.153	-0.112	-0.034	-0.034
forumnv	0.057	0.01	-0.033	-0.026	0.117	-0.01	0.152	0.114	0.936	1	0.1435	0.142	-0.03	-0.03	0.251	0.23	0.053	-0.105	-0.031	-0.031
glossary	0.254	0.105	-0.011	-0.009	0.038	-0.01	0.107	0.104	0.163	0.1435	1	0.902	-0.008	-0.008	0.307	0.272	0.022	-0.022	-0.008	-0.008
glossary nv	0.225	0.089	-0.009	-0.007	0.076	-0.01	0.204	0.187	0.146	0.1418	0.9016	1	-0.006	-0.006	0.518	0.466	0.058	0.0082	-0.006	-0.006
lesson	-0.01	-0.01	-0.004	-0.003	-0.01	-0	-0.01	-0.01	-0.032	-0.03	-0.008	-0.01	1	1	-0.01	-0	-0.008	0.1329	-0.003	-0.003
lesson nv	-0.01	-0.01	-0.004	-0.003	-0.01	-0	-0.01	-0.01	-0.032	-0.03	-0.008	-0.01	1	1	-0.01	-0	-0.008	0.1329	-0.003	-0.003
quiz	0.176	0.108	-0.006	-0.004	0.235	0.046	0.414	0.417	0.174	0.251	0.3067	0.518	-0.008	-0.008	1	0.897	0.12	0.0051	-0.008	-0.008
quiz nv	0.109	0.049	-0.005	-0.004	0.245	0.002	0.362	0.466	0.157	0.2298	0.2717	0.466	-0.005	-0.005	0.897	1	0.106	0.0208	-0.005	-0.005
survey	0.118	0.126	0.0087	0.002	0.153	0.083	0.134	0.088	0.153	0.0532	0.0219	0.058	-0.008	-0.008	0.12	0.106	1	-0.037	-0.008	-0.008
userid	-0.07	-0.04	-0.067	-0.071	-0.04	-0.02	-0.02	0.056	-0.112	-0.105	-0.022	0.008	0.133	0.1329	0.005	0.021	-0.037	1	-0.025	-0.025
wiki	-0.01	-0.01	-0.004	-0.003	-0.01	-0	-0.01	-0.01	-0.034	-0.031	-0.008	-0.01	-0.003	-0.003	-0.01	-0	-0.008	-0.025	1	1
wikinv	-0.01	-0.01	-0.004	-0.003	-0.01	-0	-0.01	-0.01	-0.034	-0.031	-0.008	-0.01	-0.003	-0.003	-0.01	-0	-0.008	-0.025	1	1

Слика 11 Матрица на корелација на атрибутите

Figure 11 Attribute Correlation Matrix

Од оваа матрицата на корелација многу лесно може да се воочи поврзаност меѓу атрибутите за приказ на активностите во модулот квиз и модулот задача. Ова би можело да послужи на пример како одреден вид на асоцијациска зависност меѓу овие атрибути, што би значело дека зголемена активност на модулот за задачи значи и зголемена активност на модулот квиз и обратно. За некои од атрибутите пак, може да се констатираат одредени негативни корелациски вредности што исклучува било каква поврзаност меѓу нив.

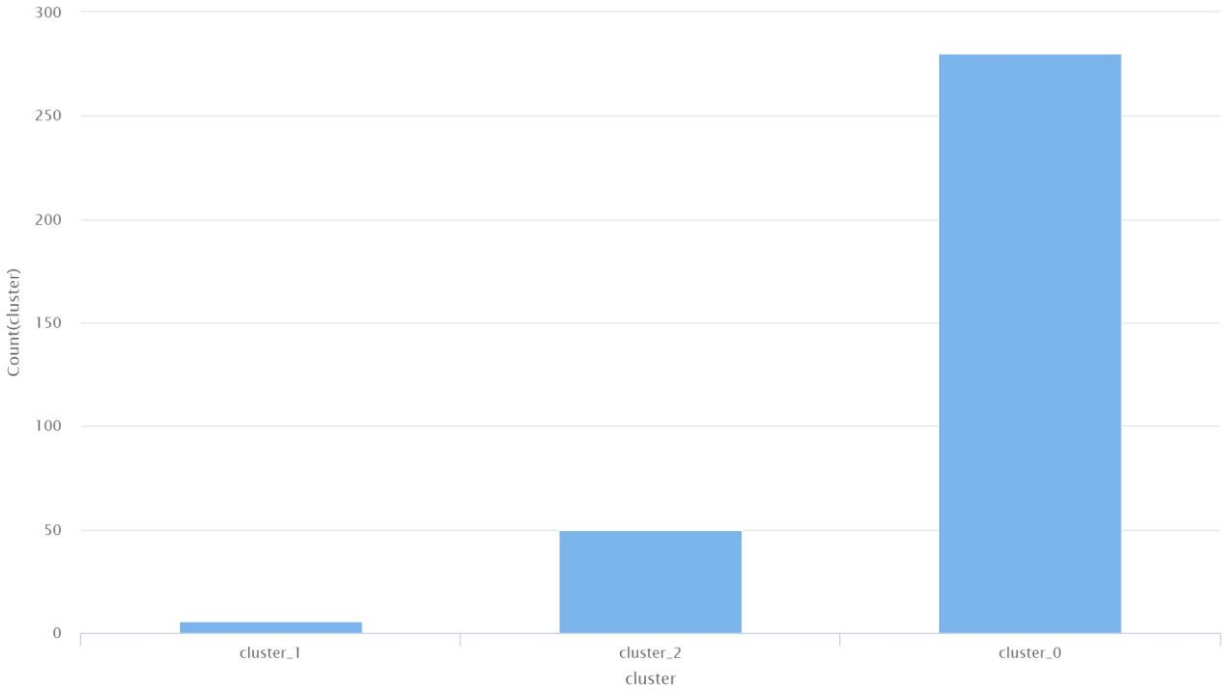
Пред да се изврши кластерирањето со помош на алгоритмот K-means, е неопходна, но не и задолжителна постапката на нормализација на податочниот сет, сè со цел да се подобрат резултатите од кластерирањето. Постојат повеќе методи за подобрување на перформансите на овој алгоритам, меѓу кои доста значајни се техниките за стандардизација на податочниот сет. Стандардизацијата е централен препроцесирачки чекор во податочното рударење со која се овозможува стандардизирање на вредностите на атрибутите од различен, односно динамичен ранг, во одреден спефицичен ранг на вредности. Нормализираниот податочен сет после извршената нормализација изгледа вака:

Row No.	userid	prediction(u...	assign	assign nv	forum	forumnv	quiz	quiz nv
1	8000	9449.185	8.976	14.079	-0.034	-0.089	1.331	0.141
2	7888	9449.185	10.819	9.924	-0.059	-0.045	-0.142	-0.085
3	10113	9447.695	0.276	0.149	-0.020	-0.057	3.779	9.362
4	5913	7603.611	-0.240	-0.141	2.675	3.019	-0.142	-0.085
5	3967	8413.271	1.398	0.111	2.929	2.730	-0.125	-0.085
6	10346	9195.272	2.509	2.171	2.305	0.486	-0.142	-0.085
7	8285	7269.280	-0.240	-0.141	2.128	2.880	-0.142	-0.085
8	14436	7201.216	-0.240	-0.141	1.983	2.852	0.008	-0.062
9	7451	7492.312	-0.240	-0.141	2.612	2.827	-0.142	-0.085
10	14727	7650.983	-0.240	-0.141	3.000	2.513	-0.142	-0.085
11	4965	7239.423	-0.230	-0.141	2.427	2.211	-0.133	-0.085
12	7020	6950.011	-0.240	-0.141	1.623	2.114	-0.142	-0.085

Слика 12 Нормализиран податочен сет

Figure 12 Normalized data set

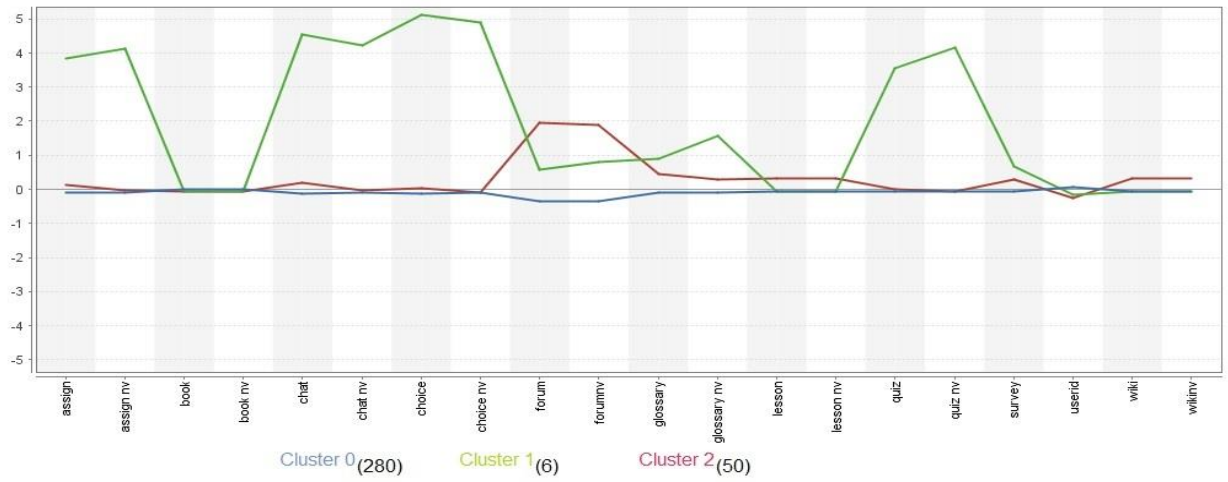
Со примена на Elbow метод е одредено дека K=3 и ако визуелизацијата се направи на 20 атрибути, од вкупно анализирани 336 наставници, бројот на наставници кој е сместен во секои од овие три кластери е Cluster 0 е 280 наставници, Cluster 1 е 6 наставници и Cluster 2 е 50 наставници.



Слика 13 Број на наставници во секој од добиените кластери

Figure 13 Number of teachers in each of the clusters obtained

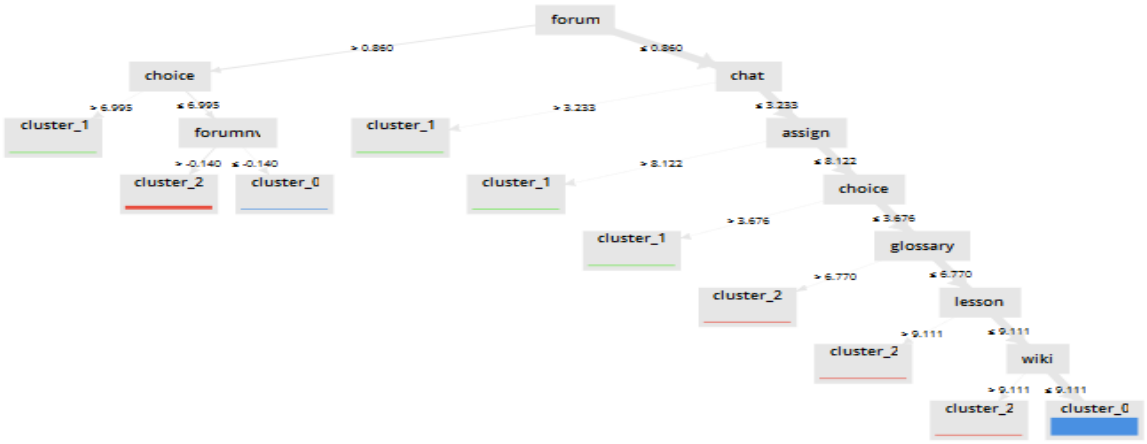
Од претставените податоци може да се заклучи дека во првиот кластер има најмалку наставници, додека пак најброен е кластерот 0. Сепак, за поконкретна анализа е потребно да утврдиме какви се активностите на наставниците во секој од кластерите, а на тој начин ќе можеме да донесеме заклучок кој од кластерите на наставници покажале најголема активност, кои најмала, а која група на наставници се наоѓаат помеѓу наведените две, врз основа на нивните активности во испитуваните модули на системот за електронско учење - Moodle. За таа цел е неопходен нов начин за визуелизација на кластерите кој ќе ни презентира подетални информации во врска со модулите и активностите на наставниците. Еве како изгледа добиениот графички приказ на атрибутите од податочното множество и добиените кластери:



Слика 14 Графикон на растојание на атрибутите од центарот на центроидите

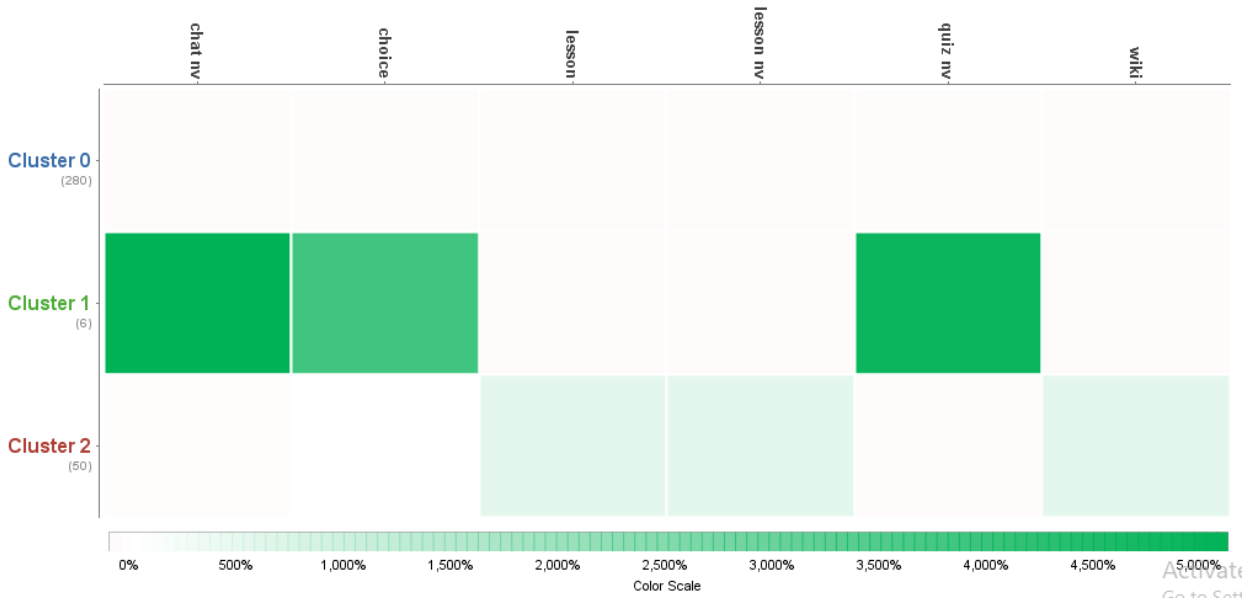
Figure 14 Graph of the distance of the centroid of the attributes

Вака добиениот графички приказ на кластерите и секоја од активностите од податочниот сет, може да се утврди дека во кластерот 0 припаѓат наставници кои покажале најголема активност во испитуваните модули. Кластерот 1 ги опфаќа наставниците кои биле најмалку активни и минимално ги користеле можностите кои ги нудат дел од модулите на Moodle. Кластерот 2 пак, е претставен од наставници кои со својата активност се наоѓаат помеѓу двата останати кластери.



Слика 15 Дрво на одлучување

Figure 15 Decision tree



Слика 16 Графикон на атрибутите во кластерите

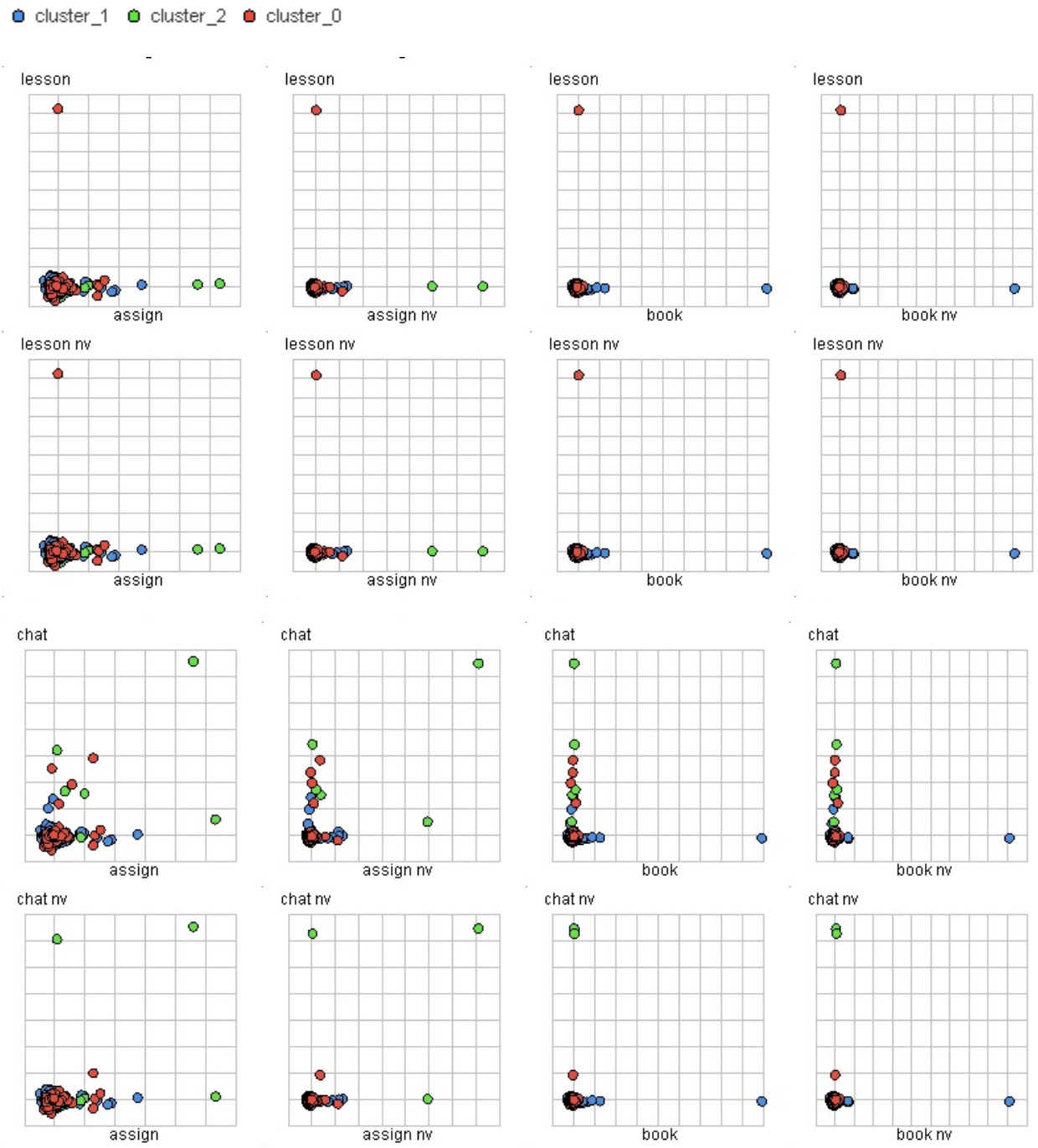
Figure 16 Graph of the attributes in the clusters

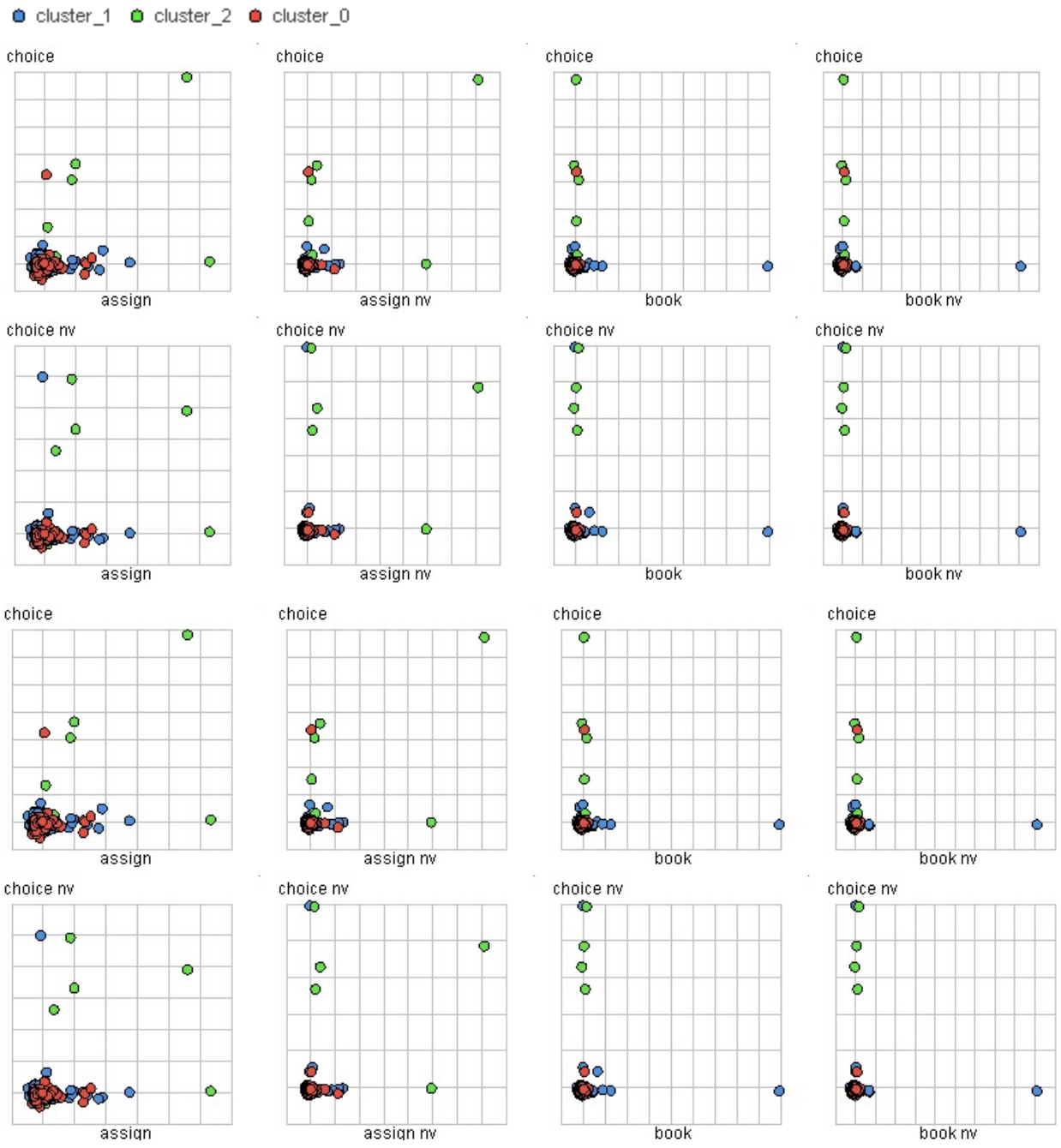
Исто така, со помош на овој приказ многу лесно може да се воочи дека од сите испитувани модули, модулот за кој наставниците покажале најголема активност е модулот квиз. Од останатите модули, за кои се забележува поголема активност на наставниците, може да се наведат модулите за лекција и чат.

6.2.4 Евалуација и анализа на добиените резултати

Целта на ова истражување е да се направи дескриптивна анализа на податоците од Moodle базата на податоци. Поконкретно, анализа на активностите на наставниците во определени модули од системот за електронско учење. Притоа, за таа цел беа користени повеќе алатки за пребарување на бази на податоци со цел да се креира добар податочен сет кој ќе биде основа за квалитетна понатамошна анализа. Со оглед на структурата и карактеристиките на податоците кои се предмет на обработка (податоци добиени од систем кој се користи за едукативни цели), следниот чекор во истражувањето беше примена на K-means алгоритамот за кластерирање. Одлуката за примена на овој вид на податочно рударење се должи на фактот што предмет на анализа е огромен податочен сет, со голем број на корисници, и оттука слободно може да се каже дека веројатноста за наоѓање на скриени релации во податоците е голема. Исто така, со оваа анализа се добива преглед на користењето на секој од испитуваните Moodle модули. Дополнително, може да се утврдат групи на наставници со заеднички карактеристики - односно во случајот на ова истражување, се добиени три групи на наставници согласно нивната активност во посочените модули во системот. Неминовно е да се напомене дека кластерирањето како вид на податочно рударење, е метода за анализа за која не постои прецизен и сигурен начин на кој би се проверила нејзината точност. Тоа значи дека не можеме да бидеме сигурни во точноста на добиените резултати, туку наместо тоа, единствено може да се бара начин за утврдување на состојбата, извлекување на одредени заклучоци и барање на релации во она што веќе е познато и она што е добиено како краен исход од анализата. Исто така, излезните резултати од примената на K-means алгоритамот, зависат од стартната поставеност на центроидите, како и од бројот на кластери кои е дефиниран. Сè со цел да се извлечат одредени значајни информации за начинот на користење на системот за електронско учење – Moodle, во продолжение е прикажан преглед на најчесто користените модули од Moodle, според информациите кои ги дава графиконот прикажан на *Слика 16*. На сликата што следува е прикажана релацијата помеѓу

овие модули, односно нивната употреба од страна на наставниците, во периодот што беше предмет на анализа:



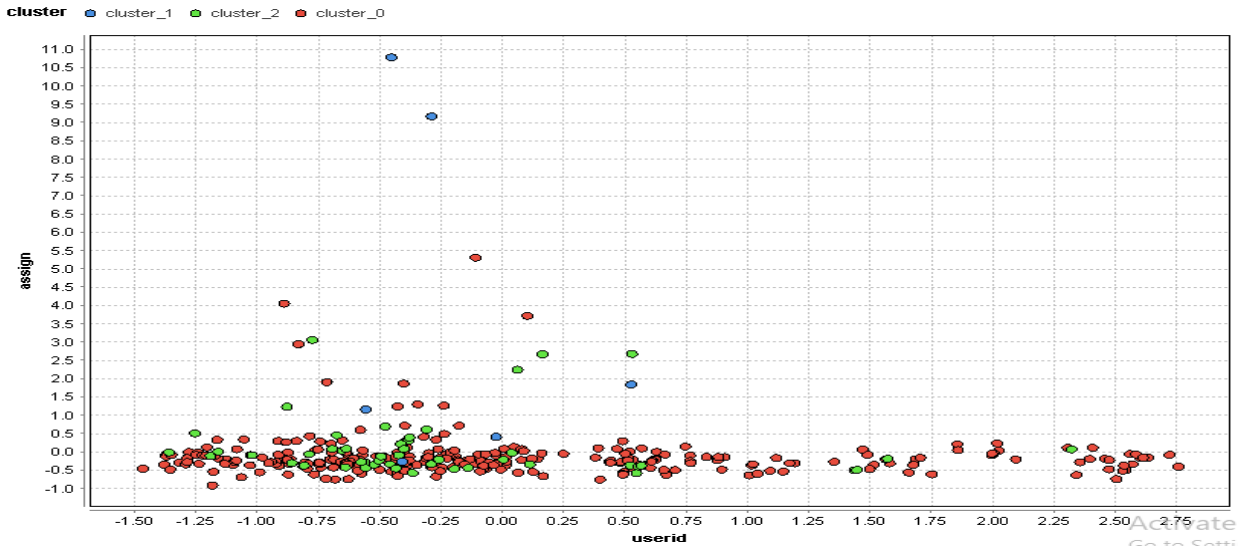


Слика 17 Приказ на корелацијата меѓу модулите: избор, книга, задача, лекција и соодветни кластери

Figure 17 View of the correlation between modules: choice, book, assign, lesson and appropriate clusters

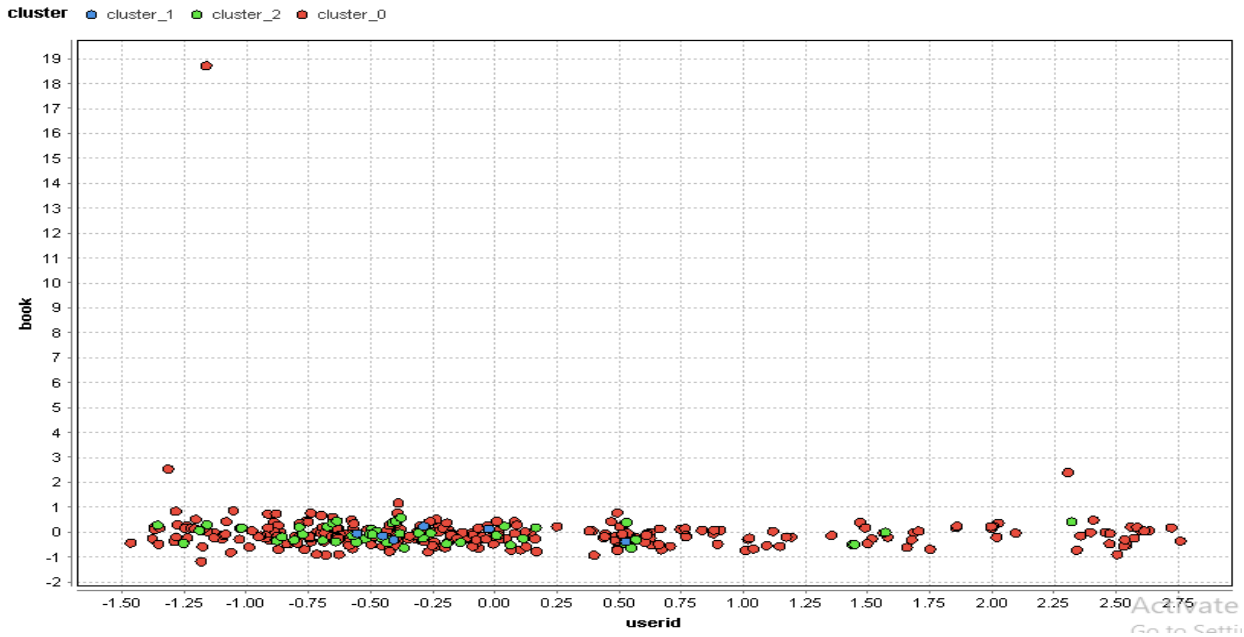
Во однос на дефинираните кластери и активностите на наставниците во овие модули, за кои е утврдено дека се едни од помалку посетуваните од страна

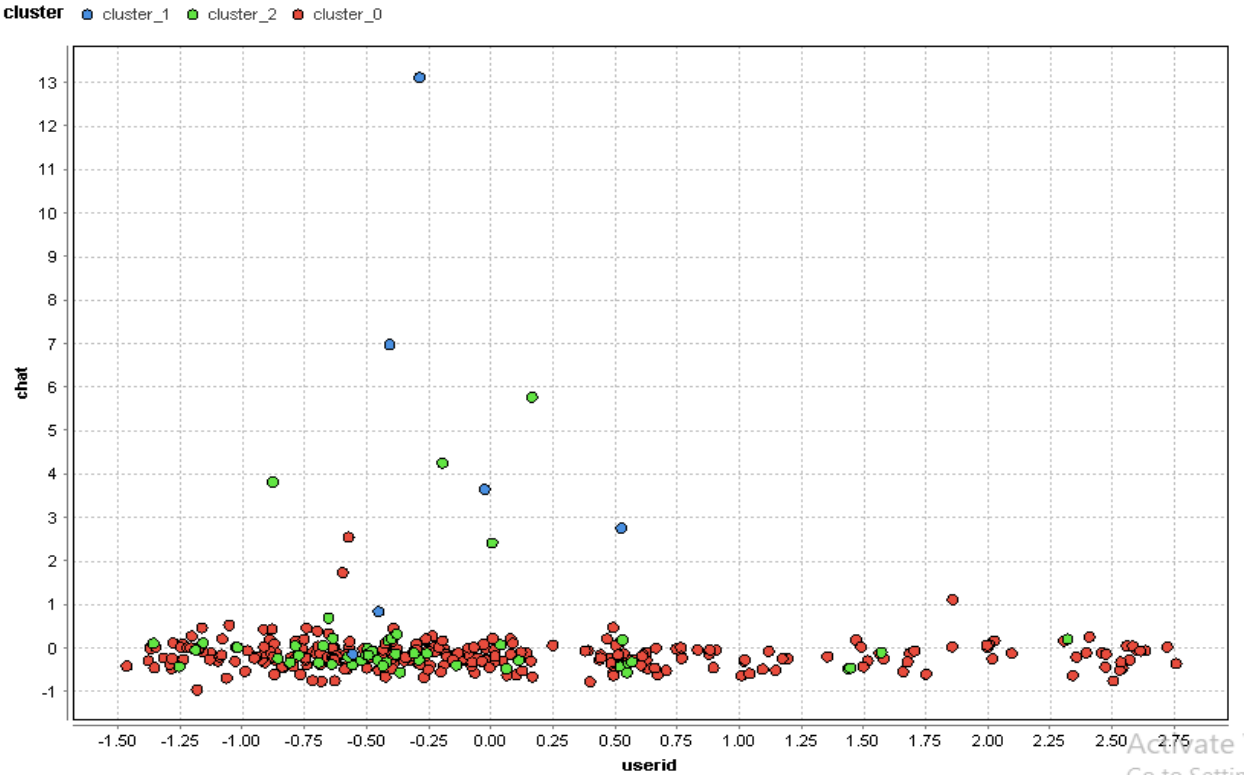
на наставниците (во периодот што е предмет на испитување), може да се забележи дека и во овој случај, онаа мала група на наставници кои покажале најголема активност и во овие модули, се сместени во кластерот 0, заедно со наставниците што покажале најмала активност во најупотребуваните модули од системот за електронско учење во кластерот 1. За кластерот 2, исто како и во претходниот случај, според анализата на корелациите меѓу најмалку користените модули, може да се заклучи дека, во овој кластер припаѓаат наставници кои врз основа на нивните активности може да се класифицираат помеѓу другите две групи, односно кластери од наставници. Направена е поделба на наставниците во три кластери, а сето тоа во зависност од нивните активности во анализираните модули, во продолжение се претставени графикони на кои е прикажана распределбата на наставниците во дефинираните кластери, за секој од испитуваните модули поединечно. За секој од прикажаните графикони, на *x*-оската се претставени корисничките имиња на наставниците (userid), а на *y*-оската е прикажан бројот на активностите на наставниците за соодветниот модул - *assign, forum, quiz, chat, choice, glossary, lesson u book*:



Слика 18 Распределба на наставниците во зависност од бројот на активности во модулот задача (assign)

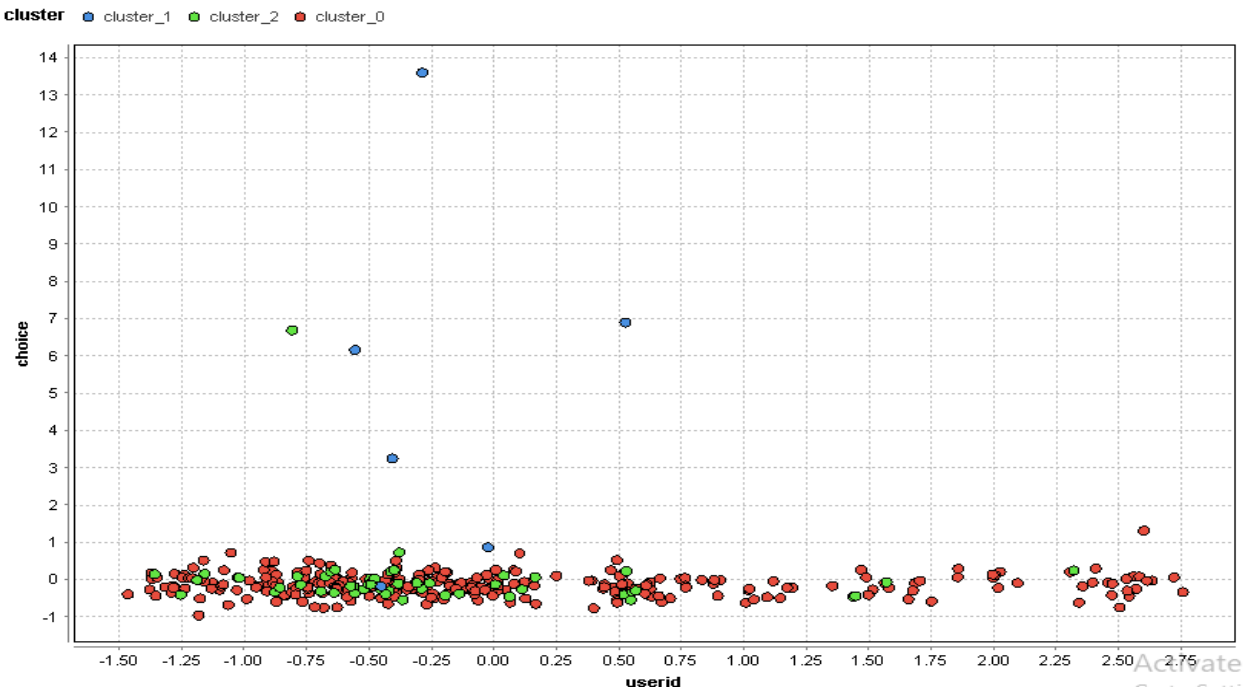
Figure 18 Distribution of teachers depending on the number of activities in the assignment module

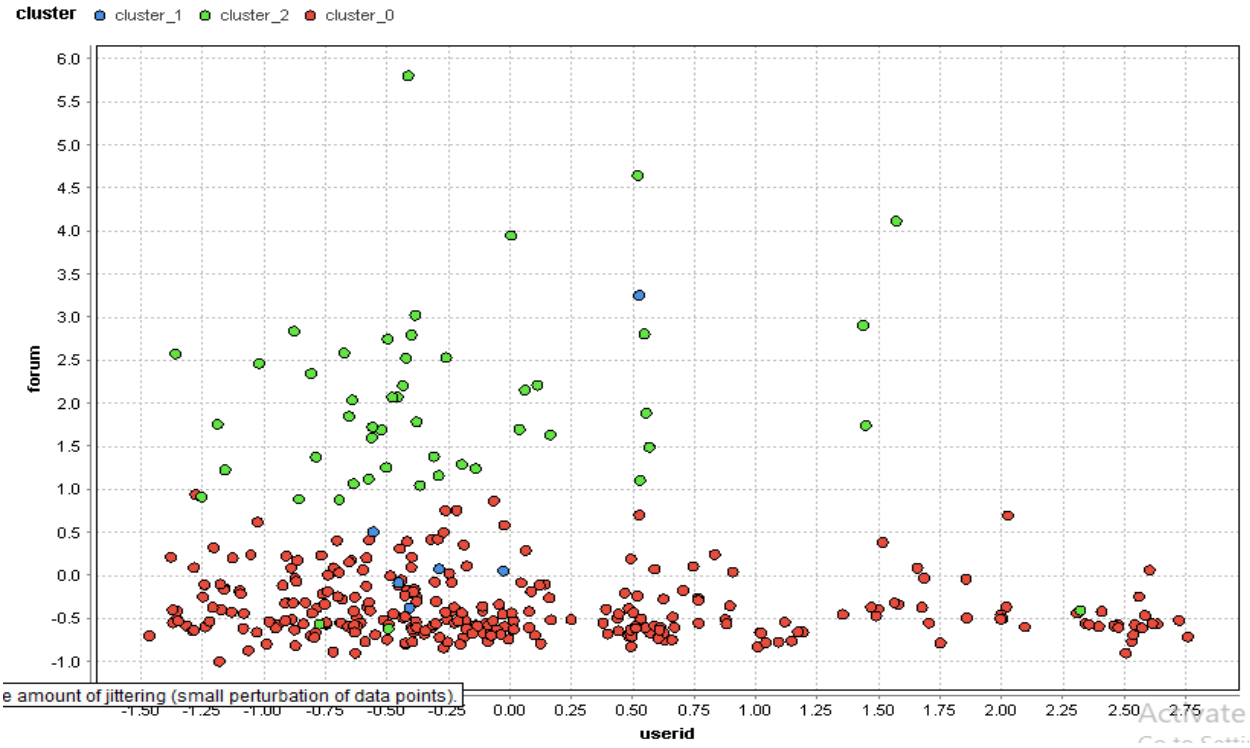




Слика 19 Распределба на наставниците во кластерите во зависност од бројот на активности во модулот книга (book) и разговор (chat)

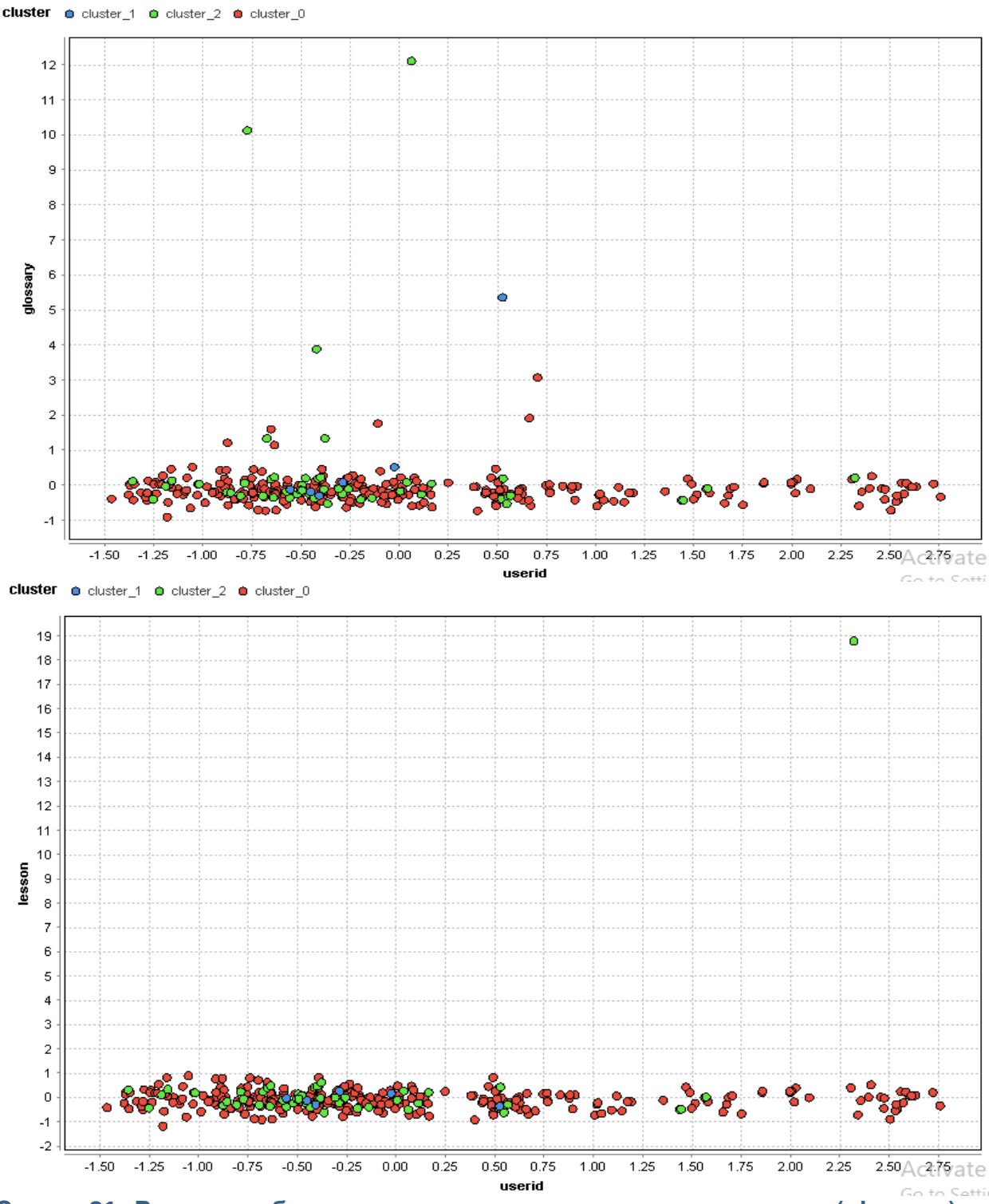
Figure 19 Distribution of cluster teachers depending on the number of activities in the book module and chat





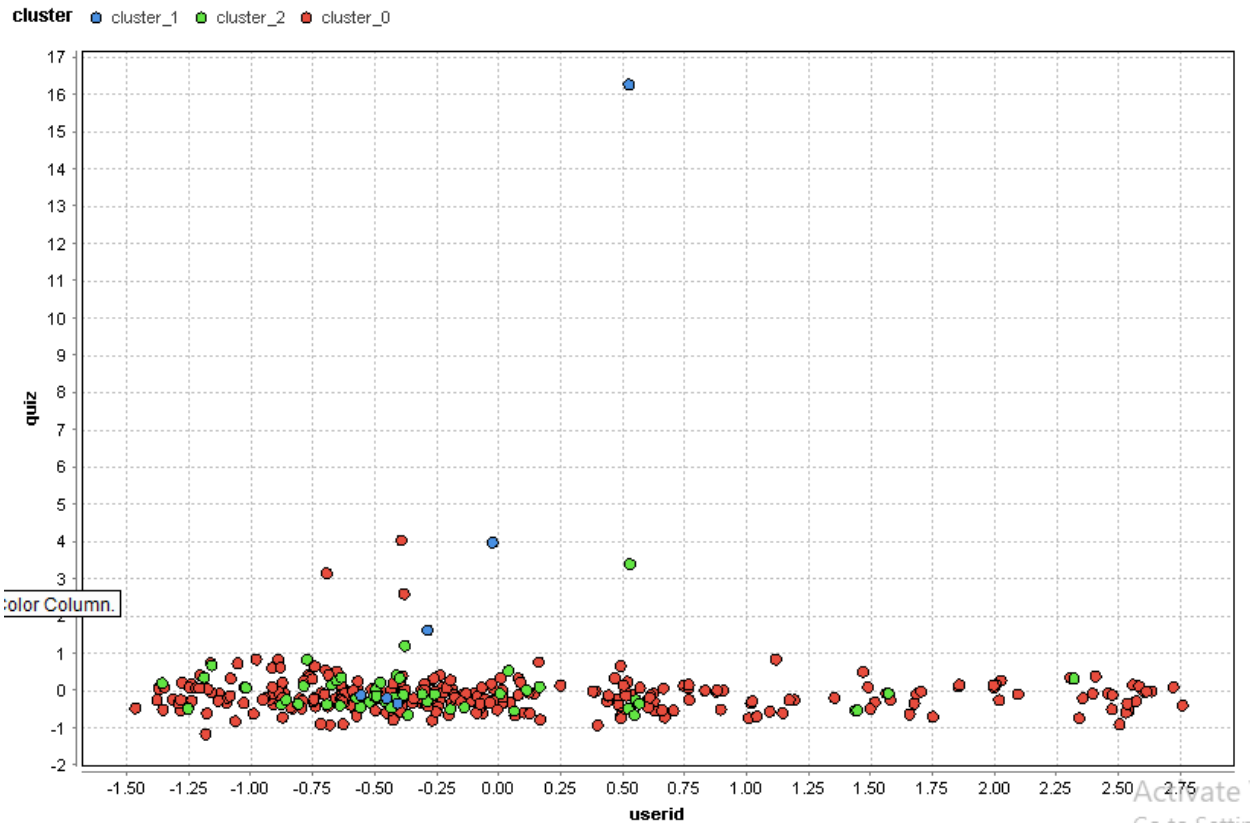
Слика 20 Распределба на наставниците во кластерите во зависност од бројот на активности во модулот избор (choice) и форум (forum)

Figure 20 Distribution of teachers in clusters depending on the number of activities in the choice module and forum



Слика 21 Распределба на наставниците во модулот речник (glossary) и лекција (lesson)

Figure 21 Teacher distribution in the glossary and lesson modules



Слика 22 Распределба на наставниците во модулот квиз (quiz)

Figure 22 Distribution of teachers in the quiz module

Врз основа на овие графици, може да се заклучи дека кога станува збор за почесто користените модули, во кластерот 1, кој содржи најмал број наставници, се наоѓаат наставници кои покажале најголем број активности и заинтересираност за испитуваните модули. Како што беше кажано и претходно, врз основа на претходно направените анализи, и со овој пристап, односно врз основа на прикажаните графикони, може да се утврди дека во најбројниот кластер - кластерот 0 се кластерирани наставниците кои генерално зборувано, имаат најмал број активности во модулите од системот за електронско учење кои беа предмет на истражување. Ова може да се забележи од графичкиот приказ на *Слика 22*, каде е претставена распределбата на наставниците во кластерите и бројот на активности во модулот задача. Според прикажаните графици, кога зборуваме за модулите кои се помалку користени, односно кои наставници покажале помала заинтересираност, споменатата распределба на наставници во

кластерите, не е така прецизна и јасна како што е случајот кај модулите за кои се забележани најголем број активности од наставниците.

Овде, може да се направи анализа на резултатите кои ги покажале наставниците во соодветниот период, за кој се однесува користењето на испитуваните модули. Од добиените резултати од кластерирањето може да се искористат како основа, односно почетна точка за реализација на друг вид анализа, како што се класификација или пак асоцијација каде што би се работело за наоѓање на одредени поврзаности внатре во утврдените кластери од наставници. На пример, како таргет група може да бидат посочени наставниците кои припаѓаат на кластерот карактеризиран со најмалку реализирани активности, со цел да се утврди причината за минималната активност на оваа група на наставници. Затоа со таа анализа се дефинираат слабостите и се бара начини за нивно отстранување или минимизирање на последиците од нив, а сето тоа со единствена цел добивање на оптимална средина за реализација на едукативните процеси и постигнување на најдобриот можен исход при ваквите активности.

7. Заклучок

Во оваа магистерска работа основната идеја беше да се направи анализа на однесувањето на наставниците во системот за електронско учење, за да се добие претстава за прогресот на наставниците во процесите на образованието. Ваквите анализи се реализираат со примена на техниките во едукативното податочно рударење – дисциплина која содржи пакет од информатички и психолошки методи за обработка на податоците генерирани од учесниците во процесите на учење. Основна цел на оваа истражувачка дисциплина е развојот на различните приоди на анализа со цел разбирање на тоа колку ги користат модулите на системот за електронско учење – Moodle, на Универзитетот „Гоце Делчев" – Штип.

Во оваа истражувачка работа беа применети алатки и технологии за обработка на големи податоци, затоа може да се заклучи дека во случај на процесирање на податоци со карактеристики на големи податоци, доста е битна количината на податоци со кои располагаме - од аспект на перформансите на упитите кои би ги користеле за обработка, и секако алатките кои се користат при процесирањето. Меѓутоа, уште повеќе е значаен методот на обработка на податоците, односно техниката на анализа бидејќи од тоа зависи какви резултати би се добиле на крај. Значи од голема важност е изборот на техниката за обработка на податоците, кој пак од своја страна треба да биде направен врз основа на видот на податоците кои се предмет на обработка, нивната структура - форматот во кој се наоѓаат како и од она што се очекувани резултати, односно исход кој треба да биде добиен на крај од извршената анализа.

Понатаму врз основа на добиените резултати од кластерирањето на податочниот сет составен од податоци добиени од активностите на наставниците на системот за електронско учење, може да се констатира состојбата во врска со најкористените модули на системот за електронско учење – Moodle, од Универзитетот „Гоце Делчев" –Штип, во испитуваниот период од 2012 г. до 2019 г. Дополнително може да се направи анализа на наставниците кои имале најголема активност, а кои најмала. Затоа може да се донесат одредени заклучоци за тоа какви мерки да бидат преземени со цел зголемување на активностите на

наставниците на системот, а со тоа и подобрување на начините на реализирање на наставата.

Притоа, ваквите анализи покажале дека наставниците кои биле најактивни и покажале најголем интерес за опциите кои ги нудат различните модули на системот за е-учење, покажале подобри резултати при реализирањето на наставата од страна на наставниците, како и при полагање на испитот од страна на студентите. На овој начин, им се дава простор и повеќе време на професорите, знаејќи каде и како да им дадат дополнителни објаснувања на студентите во врска со испитот, сè со цел да се подобрат резултатите на полагањето на испитите од страна на студентите. Резултатите од ова истражување може да се искористат како основа за дополнителни анализи, со цел понатамошно процесирање на добиените кластери и извлекување на значајни информации. Ова значи дека кластерирањето како техника, може да послужи како основа за други техники за податочно рударење, поради фактот што со добиените кластери веќе постои и е добиена одредена поврзаност, односно класификација на податоците од почетниот стадиум.

Врз основа на горенаведеното, може да се заклучи дека со обработката на податоците од системите за е-учење, се овозможува утврдување на начинот на реализација на едукативните процеси, односно се добиваат информации за професорите колку го користат системот за е-учење за непосредна комуникација со студентите. Исто така, на овој начин се добиваат информации во врска со тоа дали презентираните материјали за учење се согласно потребите и барањата на студентите, за да, врз основа на тоа, се направи прилагодување на содржината на курсевите како и методите кои се користат во едукативните активности. Дополнително, се добива приказ, односно претстава за користењето на различните модули и можности што ги нуди електронската околина за учење, а како резултат на тоа се дефинираат различностите меѓу групи од наставници, што во исто време претставува и одредување на заеднички карактеристики меѓу наставниците кои припаѓаат на иста група (кластер). Со други зборови, може да се каже дека, со реализација на вакви и слични истражувачки и аналитички активности, се добиваат информации кои може да се искористат за оптимизација

на комуникацијата меѓу студентите и професорите, со цел искористување на максималните капацитети кои тие ги поседуваат, но и целосна употреба на сѐ она што го овозможуваат системите за електронско учење кои заедно со бројните информатички достигнувања претставуваат причина за значајни промени во образовниот систем и сите негови составни компоненти.

8. Користена литература

[1] Saptarshi Ray. (2013). Big Data in Education:Gravity, the Great Lakes Magazine, pp. 8-10..

[2] Zhou, Z. H., Chawla, N. V., Jin, Y., & Williams, G. J. (2014). Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]. IEEE Computational Intelligence Magazine, 9(4), 62-74..

[3] Sunila Gollapudi.(2013).Getting Started with Greenplum for Big Data Analytics: Published by Packt Publishing Ltd. Birmingham B3 2PB, UK..

[4] S. M. S. S. H. Salisu Musa Borodo.(2016) ."Big Data Platforms and Techniques," Indonesian Journal of Electrical Engineering and Computer Science, vol. 1, pp. 191-200. .

[5] Drigas, A.S, Leliopoulos, P. (2012).The Use of Big Data in Education: IJCSI International Journal of Computer Science Issues..

[6] Chen, C.P., Zhang, C.-Y.(2014). Data-intensive applications, challenges, techniques: a survey on big data. Inf. Sci. 275, 314–347..

[7] Khan, N., Yaqoob, I., Hashem, I.A.T., Inayat, Z., Mahmoud Ali, W.K., Alam, M., Shiraz,M., Gani, A.(2014). Big data: survey, technologies, opportunities, and challenges.Sci. World J..

[8] Wang, L.(2016). Machine learning in big data. Int. J. Adv. Appl. Sci. 4, 117–123..

[9] Jadhav, A., Deshpande, L.(2016). A survey on approaches to efficient classification of data streams using concept drift. Int. J. 4..

[10] Sun, J., Fujita, H., Chen, P., Li, H.(2017). Dynamic financial distress prediction with concept drift based on time weighting combined with adaboost support vector machine ensemble. Knowledge-Based Syst. 120, 4–14..

[11] Razzak, M.I., Naz, S., Zaib, A.(2017). Deep learning for medical image processing: Overview, challenges and future. arXiv preprint arXiv:1704.06825.

[12] Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R.,Muharemagic, E.(2015). Deep learning applications and challenges in big data analytics. J. Big Data 2, 1..

[13] Zang, W., Zhang, P., Zhou, C., Guo, L.(2014). Comparative study between incremental and ensemble learning on data streams: case study. J. Big Data 1, 1–16..

[14] Skowron, A., Jankowski, A., Dutta, S.(2016). Interactive granular computing.Granular. Computing 1, 95–113..

[15] Wang, H., Xu, Z., Pedrycz, W.(2017). An overview on the roles of fuzzy set techniques in big data processing: trends, challenges and opportunities. Knowledge-BasedSyst. 118, 15–30..

[16] Huang, Y., Li, T., Luo, C., Fujita, H., Horng, S.-J.(2017). Matrix-based dynamic updating rough fuzzy approximations for data mining. Knowledge-Based Syst.119, 273–283..

[17] Luo, C., Li, T., Chen, H., Fujita, H., Yi, Z.(2016). Efficient updating of probabilistic approximations with incremental objects. Knowledge-Based Syst. 109, 71–83..

[18] A. N. D. F. H. M. K. Judith Hurwitz.(2013).Big Data for Dummies, Hoboken, New Jersey..

[19] Usha, D., Aps, A.J.(2014). A survey of Big Data processing in perspective of hadoop and mapreduce. International Journal of Current Engineering and Technology..

[20] Mall, N.N., Rana, S., et al.(2016). Overview of big data and hadoop. Imperial J.Interdiscip. Res. 2..

[21] V. Prajapati.(2013). Big Data Analytics with R and Hadoop, Birmingham B3 2PB, UK.

[22] Prasad, B.R., Agarwal, S.(2016). Comparative study of big data computing and storage tools: a review. Int. J. Database Theory App. 9, 45–66..

[23] Coronel, C., Morris, S.(2016). Database Systems: Design, Implementation, & Management. Cengage Learning..

[24] Lydia, E.L., Swarup, M.B.(2015). Big data analysis using hadoop components like flume, mapreduce, pig and hive. Int. J. Sci. Eng. Comput. Technol. 5, 390..

[25] White, T.(2012). Hadoop: The Definitive Guide. O'Reilly Media Inc...

[26] Mazumder, S.(2016). Big data tools and platforms. In: Big Data Concepts, Theories, and Applications. Springer, pp. 29–128..

[27] Loganathan, A., Sinha, A., Muthuramakrishnan, V., Natarajan, S.(2014). A systematic approach to Big Data. Int. J. Comput. Sci. Inf. Technol. 4, 869–878..

[28] Shaw, S., Vermeulen, A.F., Gupta, A., Kjerrumgaard, D.(2016). Hive architecture. In: Practical Hive. Springer, pp. 37–48..

[29] Sakr, S.(2016). Big data 2.0 processing systems: a survey. Springer Briefs in Computer Science..

[30] Beyer, K.S., Ercegovac, V., Gemulla, R., Balmin, A., Eltabakh, M., Kanne, C.-C., Ozcan,F., Shekita, E.J.(2011). Jaql: a scripting language for large scale semistructured data analysis. In: Proceedings of VLDB Conference..

[31] "Apache-Mahout" [Online]. Available: https://mapr.com/products/product-overview/apache-mahout/.

[32] Islam, M.K., Srinivasan, A.(2015). Apache Oozie: The Workflow Scheduler for Hadoop. O'Reilly Media Inc...

[33] Wadkar, S., Siddalingaiah, M.(2014). Apache Ambari. In: Pro Apache Hadoop.Springer, pp. 399–401..

[34] Sammer, E.(2012). Hadoop Operations. OÕReilly Media Inc..

[35] Chullipparambil, C.P.(2016). Big Data Analytics Using Hadoop Tools (Ph.D. thesis).San Diego State University..

[36] Shapira, G., Seidman, J., Malaska, T., Grover, M.(2015). Hadoop Application Architectures. O'Reilly Media Inc..

[37] Azarmi, B.(2016). Scalable Big Data Architecture. Springer..

[38] Kobielus, J.G.(2012). The forrester wave: Enterprise hadoop solutions,Forrester..

[39] Azarmi, B.(2016). The big (data) problem. In: Scalable Big Data Architecture.Springer, pp. 1–16..

[40] "Moodle," [Online]. Available: https://moodle.org/..

[41] Renato Cordeiro de Amorim (2016). A survey on feature weighting based K-Means.

Љупче Јаневски

АНАЛИЗА НА ПРИСТАПОТ КОН Е-УЧЕЊЕ СО ТЕХНИКИ ЗА ОБРАБОТКА НА ГОЛЕМИ ПОДАТОЦИ ОД MOODLE БАЗА НА ПОДАТОЦИ

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП