Универзитет „Св. Кирил и Методиј“ во Скопје
ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Дипломска работа

## Анализа на перформанси на NoSQL бази на податоци со користење податоци за срцеви заболувања


Ментор:

Проф. Д-р. Слободан Калајџиски

Студент:

Сандра Стојановска,
183045


Скопје, 2022

# Содржина

# 1. Апстракт

Со подемот на новите технологии, создавањето на онлајн светот и прекумерниот обем на податоци, се создаде ситуација во која информациите можат да се искористат за откривање знаење од самите податоци. Станавме свесни дека многу е важен квалитетот на податоци, а не само квантитетот. Големите податоци т.е. Big Data не освестија дека и неструктурираните податоци, без некоја специфична шема имаат големо значење. Токму поради овој факт досега широко користените релациони бази на податоци повеќе не ги задоволуваат потребите за складирање на ваквиот тип на информации и се создава потребата од користење на нов тип на бази на податоци без однапред предефинирана шема и преголеми комплицирани пребарувања т.н. неструктурирани бази на податоци.

Во оваа дипломска работа фокусот е ставен на претпроцесирање на големо податочно множество кое ги обработува клучните фактори за предизвикување на срцево заболување, анализа на податоците и врските во истото, градење на модел за предвидување на присутност на срцево заболување, како и мерење на перформансите на најпопуларните неструктурирани бази на податоци, претставници на сите четири подвидови на неструктурирани бази на податоци. Акцент е ставен на детална анализа и визуелизација на податоците, на пронаоѓање на нивна меѓусебна поврзаност, со цел да се напишат петнаесет содржајни прашалници за да се создаде цврста подлога за анализа на перформансите на четирите бази на податоци. Анализа на перформансите ќе ги опфаќа: начинот на пишување на прашалниците, нивната синтакса, интуитивност, брзината на внес на податоците, брзината на извршување на прашалниците, како и брзината на отстранување на податоците од базите на податоци, соодветно. Дополнително, над самите податоци, изграден е и модел за предвидување на присутност на срцево заболување со помош на машинско учење во програмскиот јазик Python.

*Клучни зборови: неструктурирани, податоци, бази, претпроцесирање, алгоритам, машинско учење, перформанси, клучни фактори, заболувања, срце, свет, компании*

# 2. Вовед

Сите денешни системи и технологии се создаваат за да ни го подобрат и олеснат начинот на живот, за наше добро. Сите компании независно од кој сектор се обидуваат да го подобрат корисничкото искуство и да ги задоволат нашите потреби. Живеејќи брз начин на живот, имаме потреба од добивање брзи резултати, операции, информации. Од една страна, гигантите на пазарот се соочуваат со се поголеми кориснички барања, а од друга страна, со се поголем број на информации кои што можат да се искористат за задоволување, токму, на другата страна. Овие големи податоци, обемни информации имаат неструктурирана природа и не следат ист шаблон, што предизвикува дополнителни компликации за работа и разбирање на истите. Но, иако истите се неструктурирани и на моменти хаотични, доколку создадеме хармонија во хаосот, истите тие информации можат да станат клучни за успехот на многу идеи и бизниси.

Релационите базни на податоци се користат со децении и се употребуваат речиси насекаде, во сите индустрии. Овој вид на бази на податоци нуди фиксна шема, ја подржува т.н. ACID парадигма со која се гарантираат четирите клучни својства кои дефинираат една трансакција т.е. се гарантира: атомичност, конзистентност, изолација и истрајност при извршувањето на било какви операции со помош на т.н. SQL јазик. Токму овие карактеристики ги прават релационите бази на податоци да бидат вистински избор за апликациите во кои точноста, сигурноста и конзистентноста на податоците е најважна. Такви системи се имплементирани во скоро сите сфери на нашето живеење: во медицината, банкарството, трговијата, ИТ индустријата, итн.

Од друга страна, неструктурираните бази на податоци се истакнуваат со својата едноставност за користење, брзина, достапност , еластичност и приспособливост. Поради способноста за брз одговор на барања кои одземаат многу време кај релационите бази на податоци при обработка на обемни податоци, овие бази на податоци стануваат се популарни и широко користени. Тие се водат според т.н. CAP теорема. Оваа теорема истакнува дека дистрибуираните податочни системи ќе понудат компромис помеѓу конзистентноста, достапноста и толеранцијата на партициите, и дека секоја база на податоци може да гарантира само две од овие три својства.

Самото широко познато и применувано англиско име на неструктурираните бази на податоци е доста дескриптивно само по себе т.н. NoSQL значи „Not only SQL “. Овие бази на податоци ги надминуваат сите пречки кои ги имаат релационите бази на податоци за справување со полуструктурираните и неструктурираните обемни податоци и се навистина моќна алатка на денешницата. Неструктурираните бази на податоци добро познатите редици и табели кои ги нудат релационите бази на податоци ги заменуваат со: JSON

објекти, клуч вредност парови, јазли и врски или колони. Постојат четири подвидови на неструктурирани бази на податоци така наречени: key value, column-oriented, document-oriented, graph based. Главна цел на овој дипломски труд е да ги спореди перформансите на по еден претставник од сите овие подвидови на неструктурирани бази на податоци. Во продолжение ќе се запознаеме со Amazon DynamoDB како претставник на неструктурираните бази на податоци базирани на клуч-вредност базите на податоци, Apache Cassandra како претставник на неструктурираните бази на податоци базирани на колони, MongoDB како неструктурирана база на податоци базирана на документи и Neo4j како граф неструктурирана база на податоци.

За детална анализа на перформансите на овие четири претставници од неструктурираните бази на податоци ќе користиме множество кое ги обработува клучните предизвикувачи на срцеви заболувања. Податочното множество кое се провлекува низ сите пори на овој дипломски труд обработува еден многу важен аспект од нашето живеење т.е. индиректно ги обработува штетните влијанија врз здравствената состојба на еден човек, како што се: брзиот начин на живот, недостатокот на квалитетен сон, уживањето во нездрава храна и нездрави пороци, стресот, физичката неактивност. Дел од овие фактори може да бидат и главни предизвикатели на тешки болести. Токму во првиот чекор на овој дипломски труд детално ќе го обработиме податочното множество. Со цел да создадеме содржајни прашалници и да ја разбереме важноста на познавањето на врските помеѓу атрибутите од ова податочно множество и нивниот квалитет, најпрво, ќе направиме претпроцесирање на податоците и детална анализа и визуелизација на податочните типови и врски. Големиот број на дијаграми, графикони, хистограми кои се дел од овој процес ќе ни овозможат добивање на претстава за структурата на податоците, нивните вредности, важноста на корелациите помеѓу атрибутите, како и за нивниот квалитет. Исто така, претпроцесирањето ќе биде клучно и за подобрување на квалитетот на податоците. Претпроцесирањето и деталната анализа ќе ни помогнат при главната цел на овој дипломски труд, мерење на перформансите на неструктурираните бази на податоци, но, ќе бидат клучни и за создавање на моделот на предвидување на присутност на срцево заболување со помош на машинско учење.

# 3. Методологија

Ова истражување се состои од неколку делови и тоа се: опис на податочното множество, претпроцесирање на податоците, дефинирање и креирање на агрегатни модели, внес и извршување на прашници во избрани четири претставници на неструктурирани бази на податоци, споредба на перформансите на различните претставници од неструктурираните бази на податоци и креирање на модел за предвидување на присутност на срцево заболување со помош на машинско учење.

Како што е и претставено на слика 1, најпрво ќе го обработиме податочното множество со цел да се запознаеме со клучните фактори кои можат да предизвикаат срцево заболување.

Во следниот чекор, ќе ги анализираме податочните типови и вредности на секој од атрибутите во податочното множество. Потоа, ќе пристапиме кон анализа на празни и невалидни вредности и на крај детално ќе ги разгледаме врските и релациите кои ги имаат атрибутите меѓусебно. Целата анализа ќе ја направиме со помош на програмскиот јазик Python, користејќи библиотеки за машинско учење.

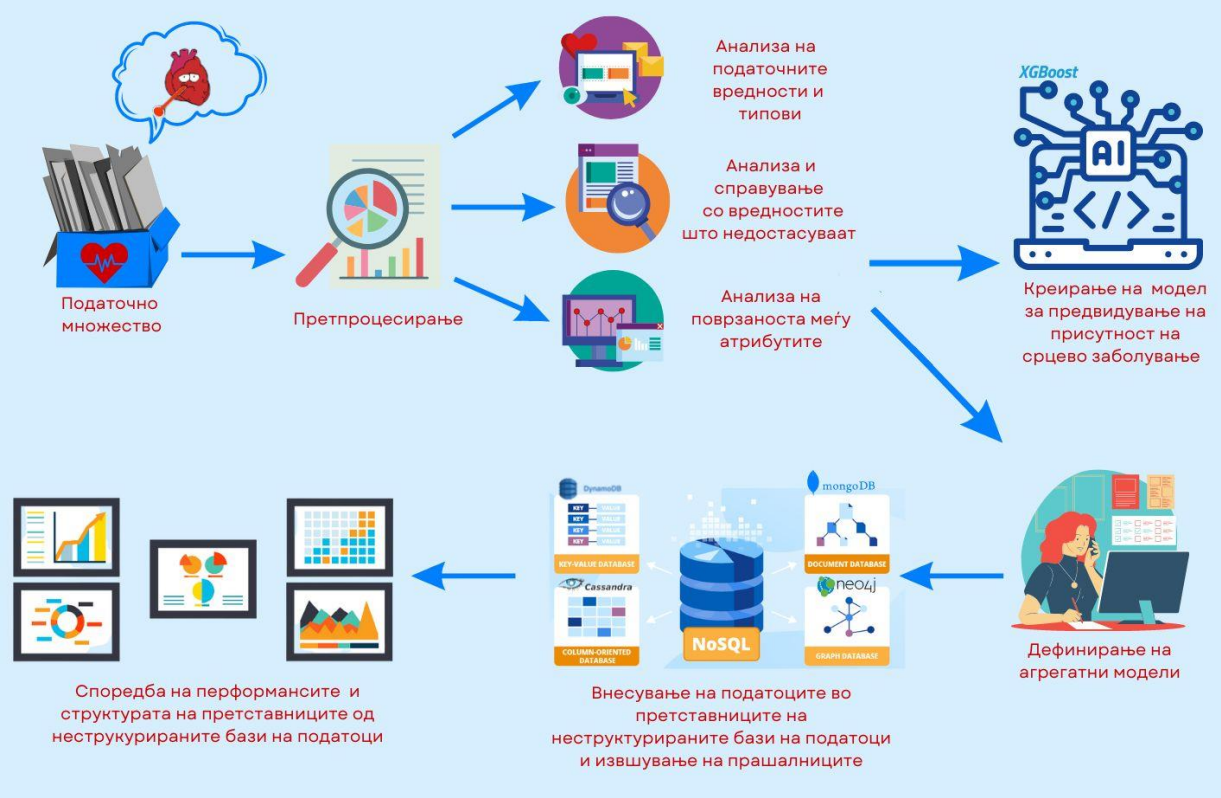
Неструктурираните бази на податоци се истакнуваат со својата едноставност за користење, брзина, достапност, еластичност и приспособливост. Токму поради овие неколку карактеристики во продолжение на овој дипломски труд ќе ги споредуваме најпопуларните и најкористените претставници од секој од видовите на неструктурирани бази на податоци и тоа: AWS DynamoDB како претставник на клуч вредност базите на податоци, Apache Cassandra како претставник на неструктурираните бази на податоци базирани на колони, MongoDB како претставник на документ базите на податоци и Neo4j како претставник на граф базите на податоци. Нашето внимание ќе биде насочено главно кон пишување на содржајни прашалници кои ќе опфатат различни агрегатни модели како групирање, минимална вредност, максимална вредност, средна т.е. просечна вредност, сума, број на записи итн.

По соодветно внесување на податоците користејќи различни техники опишани во соодветниот дел, ќе пристапиме кон извршување на 15 прашалници со цел да можеме да ја анализираме синтаксата, начинот на пишување, интуитивноста и брзината на извршување на прашалниците во секој од избраните бази на податоци. Врз основа на добиените резултати ќе направиме споредба на перформансите на по еден претставник од сите различни типови на неструктурирани бази на податоци.

За крај ќе направиме и модел за предвидување на вредноста на атрибутот срцево заболување. Модел со чија помош ќе можеме да предвидуваме според вредностите за

клучните фактори за срцево заболување дали анализираната индивидуа има или нема ваков тип на заболување.

Методологијата на истражувањето во оваа дипломска работа е дадена на слика 1.




**Слика 1 „Струкура на истражувањето“**

# 4. Опис на податочното множество

Здравјето е најважното нешто во човековиот живот. Како што вели Денис Вејтли, цитирам „Времето и здравјето се двете најскапоцени добра кои не ги препознаваме и цениме се додека не се исцрпат“. Водејќи брз начин на живот, прекумерната посветеност на кариерата и секојдневната трка со времето понекогаш не тераат да не се грижиме за себе и да ги занемаруваме знаците кои што ни ги дава нашето тело. Човековиот живот е бесценет, но, за жал, според светската здравствена организација, кардиоваскуларните заболувања вклучувајќи ги и срцевите заболувања, се главните причинители за згаснување на еден живот, глобално, одземајќи околу 17.9 милиони животи секоја година. Како главни причинители на појавата на срцево заболување се нездравата исхрана, прекумерната тежина, физичката неактивност, пушењето цигари, проблемите со менталното здравје и прекумерната употреба на алкохол.

Поради горенаведените факти истражувањето во оваа дипломска работа обработува податочно множество кое ги опфаќа главните причинители на срцево заболување. Ова податочно множество е составено од 319.795 испитаници, и има осумнаесет атрибути т.е. карактеристики.

За да се создаде ова податочно множество испитаниците, со место на живеење во Америка, одговарале на телефонски прашалници со цел да се соберат податоци за :

- Нивната телесна маса,
- Дали се класифицирале како пушачи т.е. дали испушиле над 100 цигари во нивниот живот,
- Дали истите пијат редовно алкохол т.е. дали консумираат над 14 пијалаци доколку се мажи или 7 пијалаци доколку се жени во текот на една недела,
- Дали имале мозочен удар ,
- Какво е нивното физичко здравје т.е. колку дена во месецот имале некакви проблеми со истото,
- Колку дена месечно нивното ментално здравје било нарушено,
- Дали имаат потешкотии при движење или искачување на скали,
- Нивниот пол,
- Нивната возраст, поделена во 14 категории,
- Раса,
- Дали се дијабетичари,

- Дали се физички активни,
- Какво е нивното генерално здравје,
- Колку вкупно часа, просечно, спијат во текот на еден ден,
- Дали имаат некакво заболување на бубрезите,
- Дали боледуваат од астма,
- Дали имаат рак на кожата,
- И дали имаат некој вид на срцево заболување т.е. дали имале коронарна срцева болест или миокарден инфаркт,

Врз ова богато податочно множество се извршени сите мерења на перформансите на претставниците од сите типови на неструктурирани бази на податоци. Но, најпрво е направено претпроцесирање и визуелизација на податоците сè со цел да се увидат врските и корелациите во податочното множество и детално да се разберат податоците, а воедно и да се исфилтрираат самите податоци за поедноставно, објективно мерење на перформансите на неструктурираните бази на податоци.

## 4.1. Претпроцесирање и визуелизација на податочното множество за клучни предизвикувачи на срцево заболување

### Анализа на податочните вредности и типови на секој од атрибутите

Прв чекор од ваквата анализа е вчитување на податочното множеството за понатамошна обработка, користејќи го pandas модулот во програмскиот јазик Python. Ова ни овозможува да направиме увид во структурата на множеството и да добиеме претстава за карактеристиките со кои работиме и кои ни се во фокусот на ова истражување. Дел од множеството е прикажан на сликата број 2, која следува во продолжение.

| | HeartDisease | BMI | Smoking | AlcoholDrinking | Stroke | PhysicalHealth | MentalHealth | DiffWalking | Sex | AgeCategory | Race | Diabetic | PhysicalActivity | GenHealth | SleepTime | Asthma | KidneyDisease | SkinCancer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | No | 16.60 | Yes | No | No | 3.0 | 30.0 | No | Female | 55-59 | White | Yes | Yes | Very good | 5.0 | Yes | No | Yes |
| 1 | No | 20.34 | No | No | Yes | 0.0 | 0.0 | No | Female | 80 or older | White | No | Yes | Very good | 7.0 | No | No | No |
| 2 | No | 26.58 | Yes | No | No | 20.0 | 30.0 | No | Male | 65-69 | White | Yes | Yes | Fair | 8.0 | Yes | No | No |
| 3 | No | 24.21 | No | No | No | 0.0 | 0.0 | No | Female | 75-79 | White | No | No | Good | 6.0 | No | No | Yes |
| 4 | No | 23.71 | No | No | No | 28.0 | 0.0 | Yes | Female | 40-44 | White | No | Yes | Very good | 8.0 | No | No | No |

**Слика 2 „Изглед на првите пет редови од податочното множество“**

Следен чекор во оваа анализа е да се увиди податочниот тип на секоја од карактеристиките. Овој чекор е доста значаен, бидејќи дел од карактеристиките се претставени како објекти, а истите имаат само две уникатни вредности и всушност се текстуални т.е. стринг вредности кои можат да се претворат во нумерички вредности по визуелизацијата на истите, со цел да може да се увидат меѓусебни врски меѓу податоците со помош на матрицата на

корелација и да се создаде моделот за предвидување. Приказот на податочните типови е претставен на слика 3.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 319795 entries, 0 to 319794
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   HeartDisease      319795 non-null  object
 1   BMI               319795 non-null  float64
 2   Smoking           319795 non-null  object
 3   AlcoholDrinking   319795 non-null  object
 4   Stroke            319795 non-null  object
 5   PhysicalHealth    319795 non-null  float64
 6   MentalHealth      319795 non-null  float64
 7   DiffWalking       319795 non-null  object
 8   Sex               319795 non-null  object
 9   AgeCategory       319795 non-null  object
 10  Race              319795 non-null  object
 11  Diabetic          319795 non-null  object
 12  PhysicalActivity  319795 non-null  object
 13  GenHealth         319795 non-null  object
 14  SleepTime         319795 non-null  float64
 15  Asthma            319795 non-null  object
 16  KidneyDisease     319795 non-null  object
 17  SkinCancer        319795 non-null  object
dtypes: float64(4), object(14)
memory usage: 43.9+ MB
```

**Слика 3 „Приказ на податочните типови на секоја од карактеристиките во податочното множество“**

За да ни стане појасно колку уникатни вредности има секоја од карактеристиките можеме да се осврнеме на слика 4, на која е претставен бројот на уникатни вредности за секоја колона т.е. карактеристика. Од овој приказ можеме да увидиме дека карактеристиките кои на слика 3 претставуваат објекти имаат значително помал број на уникатни вредности наспроти карактеристиките кои имаат нумерички вредности. За да може детално да ја увидиме содржината на карактеристиките со податочен тип објект, ја издвоив уникатната содржина на истите т.е. на слика 5 се дадени уникатните вредности за карактеристиките: Heart Disease, Smoking, Alcohol Drinking, Difficulty Walking, Sex, Race, Age Category, Diabetic, Physical Activity, General Health, Asthma, Kidney Disease, Skin Cancer.

```
HeartDisease          2
BMI                3604
Smoking               2
AlcoholDrinking       2
Stroke                2
PhysicalHealth       31
MentalHealth         31
DiffWalking           2
Sex                   2
AgeCategory          13
Race                  6
Diabetic              4
PhysicalActivity      2
GenHealth             5
SleepTime            24
Asthma                2
KidneyDisease         2
SkinCancer            2
dtype: int64
```

**Слика 4 „Приказ на уникатни вредности за секоја од карактеристиките во податочното множество“**

```
Unikatni vrednosti vo kolonata HeartDisease: ['No' 'Yes']
Unikatni vrednosti vo kolonata Smoking: ['Yes' 'No']
Unikatni vrednosti vo kolonata AlcoholDrinking: ['No' 'Yes']
Unikatni vrednosti vo kolonata Stroke: ['No' 'Yes']
Unikatni vrednosti vo kolonata DiffWalking: ['No' 'Yes']
Unikatni vrednosti vo kolonata Sex: ['Female' 'Male']
Unikatni vrednosti vo kolonata AgeCategory: ['55-59' '80 or older' '65-69' '75-79' '40-44' '70-74' '60-64' '50-54'
 '45-49' '18-24' '35-39' '30-34' '25-29']
Unikatni vrednosti vo kolonata Race: ['White' 'Black' 'Asian' 'American Indian/Alaskan Native' 'Other'
 'Hispanic']
Unikatni vrednosti vo kolonata Diabetic: ['Yes' 'No' 'No, borderline diabetes' 'Yes (during pregnancy)']
Unikatni vrednosti vo kolonata PhysicalActivity: ['Yes' 'No']
Unikatni vrednosti vo kolonata GenHealth: ['Very good' 'Fair' 'Good' 'Poor' 'Excellent']
Unikatni vrednosti vo kolonata Asthma: ['Yes' 'No']
Unikatni vrednosti vo kolonata KidneyDisease: ['No' 'Yes']
Unikatni vrednosti vo kolonata SkinCancer: ['Yes' 'No']
```

**Слика 5 „Уникатни вредности на карактеристиките со податочен тип - објект“**

Според информациите кои можеме да ги воочиме на слика 5, карактеристиките: Heart Disease, Smoking, Alcohol Drinking, Difficulty Walking, Sex, Physical Activity, Asthma, Kidney Disease, Skin Cancer во понатамошното претпроцесирање можат да се претворат во бројни вредности, земајќи дека „No“ и „Female“ имаат нумеричка вредност 0, а „Yes“ и „Male“ нумеричка вредност 1. За карактеристиките Diabetic и General Health самите вредности може да се генерализираат, земајќи во предвид дека ако сте имале дијабетес во текот на бременоста сепак сте имале некаков вид на дијабетес, или пак доколку сте биле на граница на развивање на оваа болест, а сепак сте го надминале тој стадиум значи дека немате дијабетес во моментот на спроведување на истражувањето и соодветно доколку вашето здравје сте го оцениле со „Poor“ и „Fair“ би добиле нумеричка вредност 0, „Very good“ и „Excellent“ би добиле нумеричка вредност 2, а „Good“ би се отсликал во нумеричка вредност 1.

### Анализа на карактеристиките со категориски вредности

За да ги разбереме детално вредностите кои ги имаат категориските карактеристики беа создадени хистограми. Во продолжение на слика 6, графички се претставени сите 14 категориски карактеристики. На истата можеме да воочиме дека е претставен и соодносот на сите вредности според категорија.


**Слика 6 „Визуелен приказ на хистограми на сите категориски карактеристики во податочното множество“**

Според податоците добиени на слика 6.1 можеме да увидиме дека многу поголем дел од испитаниците немаат срцево заболување т.е. точно 91% од испитаниците одговориле негативно на прашањето дали имаат некаков вид на срцево заболување, а само 9% дале позитивен одговор.

Но, од друга страна пак на прашањето дали испитаниците пушат или пушеле цигари имаме помало отстапување во одговорите т.е. 59% од испитаниците пушеле или пушат цигари, а 41% не се пушачи т.е. никогаш не испуштиле над 5 кутии цигари. Ова можеме да го увидиме на дел 2 од слика 6.

Евидентна е разликата во одговорите и на прашањето дали испитаниците редовно консумираат алкохол, т.е. дури 93% одговориле негативно, а само 7% се изјасниле со позитивен одговор на ова прашање. Ова можеме да го увидиме на хистограм број 3 на слика 6.

За среќа, како и во претходниот параграф , според хистограм број 4, на прашањето дали испитаниците имале мозочен удар, дури 96% од одговорите се негативни, а само 4% се позитивни.

Од хистограм број 5, прикажан на слика 6, можеме да увидиме дека потешкотии во движењето имаат само 14% од испитаниците.

За да може да се донесат објективни заклучоци за ова податочно множество скоро рамномерна е распределеноста на бројот на испитаници според нивниот пол т.е. 52% од испитаниците се мажи, а 48% од испитаниците се жени. Ова може да се увиди на хистограм број 6, од слика 6. Воедно, на хистограм број 7, можеме да заклучиме дека во податочното множество опфатени се сите 14 возрасни категории. Поголем број на испитаници имаме на возраст од 65 до 69 години т.е. 11% , 10% од испитаниците се на возраст на возраст од 60 до 64 години и од 70 до 74 години, 9% од испитаниците се на возраст од 55 до 59 години, нешто помалку т.е. 8% се на возраст од 50 до 54 години и на возраст од 80 години или постари, 7% од испитаниците се на возраст од 18-24, од 45 до 49 години и на возраст од 75 до 79 години, 6% се на возраст од 30 до 34, 35 до 39 години и од 40 до 44 години и само 5% од испитаниците се на возраст од 25 до 29 години.

Најголем број од испитаниците според раса се белци т.е. 78%, со шпанско потекло се класифицирале 8%, црнци имаме 7%, со други потекла имаме 3%, Азијци имаме 2% и американски Индијци имаме 2%. Хистограм број 8 ја дава оваа категоризација.

За да видиме дали испитаниците имаат и други болести, можеме да се фокусираме на хистограмите со реден број 9, 12, 13,14, соодветно од слика 6. Од хистограм број 9 можеме да увидиме дека 86% од испитаниците немале дијабетес или биле на граница за да добијат дијабетес но не се разболеле, а 14% имале дијабетес или добиле дијабетес за време на бременоста. Дури 87% од испитаниците немаат астма, а само 13% се соочуваат со оваа болест. Ова е евидентно на хистограм број 12. Од хистограм број 13 можеме да заклучиме дека 96% од испитаниците немаат заболување на бубрезите и 91% од испитаниците немаат ниту рак на кожата, што е воочливо на хистограм број 14.

Тесно поврзани се и хистограмите со редни броеви 10 и 11. Како физички активни во текот на 30 дена се изјасниле 78% од испитаниците, што е евидентно од хистограм број 10, а своето генерално здравје го оцениле како одлично дури 21% од испитаниците, како многу добро 36% од испитаниците, а како добро 29% од испитаниците. Воочливо на хистограм број 11 е и дека како лошо своето генерално здравје го оцениле 6%, и дури 11% како незадоволително.

### Анализа на карактеристиките со нумерички вредности

Во податочното множество освен карактеристиките со категориски вредности имаме и карактеристики со нумерички вредности. Можеме да пресметаме одредени вредности како минимална вредност, максимална вредност, просечна т.е. средна вредност, стандардна девијација, како и перцентил за да добиеме претстава за вредностите кои ги поседуваат овие карактеристики.

Според табелата, прикажана на слика 7, индексот на телесна маса (BMI) нема празни т.е. нул вредности и средната вредност изнесува 28.325399. Минималната вредност изнесува 12.02 што ни кажува дека примерокот со оваа вредност на индекс на телесна маса е под границата за здрава телесна маса која се движи во опсегот од 18.5 до 24.9. Додека пак примерокот кој го има максималниот индекс на телесна маса од 94.85 е во рангот на нездрава прекумерна телесна маса. Стандардната девијација која изразува колку членовите на групата се разликуваат од средната вредност на групата изнесува 6.3561, што ни кажува дека вредностите на примероците се распределени околу средната вредност, ова е јасно видливо доколку го погледнеме графикот број 1 на слика 8. Според перцентилите прикажани во табелата за индексот на телесна маса можеме да заклучиме дека 75% од испитаниците имаат индекс на телесна маса под 31.42, што ни кажува дека истите претежно имаат индекс на телесна маса над, во границите на нормалата телесна маса или под истата.

Од карактеристиката што го опишува физичкото здравје на испитаниците, т.е. колку дена во триесет дневен опсег истите имале потешкотии или повреди кои им го нарушиле физичкото здравје, можеме да заклучиме дека повторно немаме празни вредности и дека средната вредност изнесува 3.371710. Стандардната девијација изнесува 7.950850 и ни кажува дека податоците се распространети и подалеку од средната вредност, т.е. дека имаме варијација во вредностите за оваа карактеристика. Според вредноста на перцентилите можеме да заклучиме дека дури 75% од испитаниците се изјасниле дека во текот на триесет дена само 2 дена или помалку имале проблеми со физичкото здравје, но сепак според максималната вредност имало и испитаници кои се изјасниле дека секој ден имале нарушено физичко здравје.

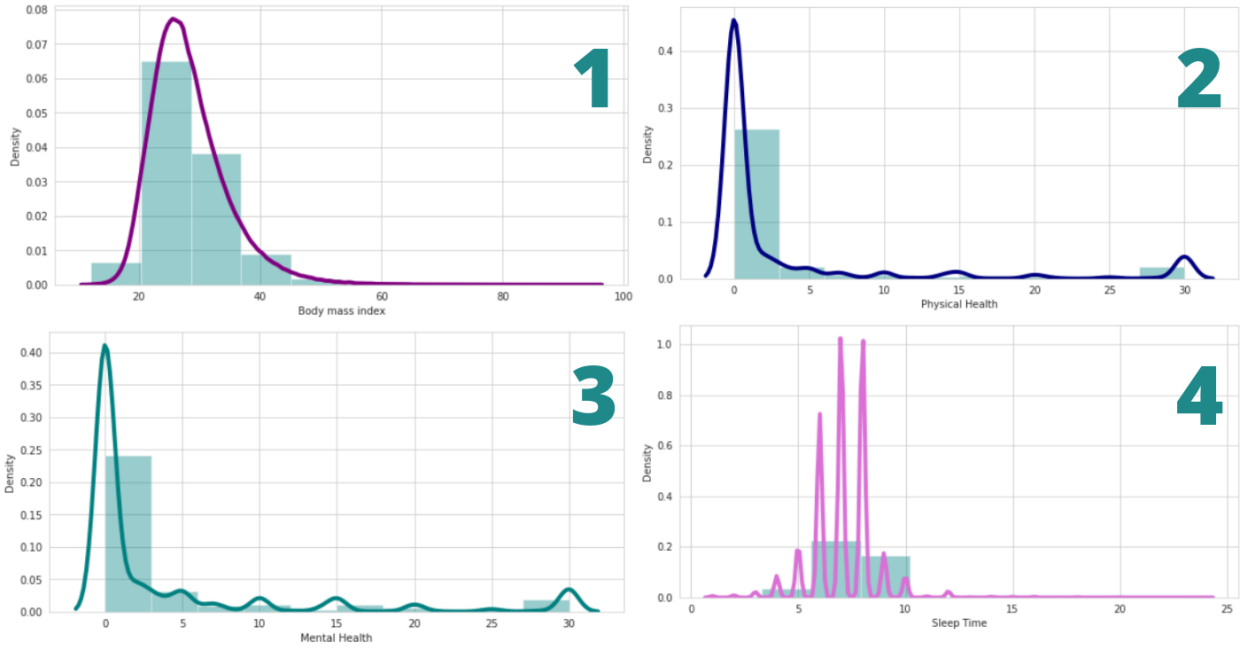
За карактеристиката која го отсликува менталното здравје, добиено е дека средната вредност изнесува 3.898366 и дека стандардната девијација изнесува 7.955235, што ни дава до знаење дека имаме варијација во податоците. Оваа карактеристика нема празни вредности и имаме испитаници што немале никакви проблеми со менталното здравје, но имаме испитаници што се соочувале секој ден во триесет дневната рамка со проблеми, ова можеме да го заклучиме од минималната и максималната вредност. За среќа, 75% од испитаниците само 3 дена или помалку имале потешкотии со менталното здравје.

Здравиот сон е еден од најбитните фактори за здрав живот, токму оваа карактеристика во моето податочно множество нема празни вредности и средната вредност изнесува 7.097075, а стандардната девијација изнесува 1.436007, овие податоци ни кажуваат дека претежно испитаниците спијат околу 7 часа дневно, што е и препорачливо време за дневен одмор и воедно од стандардната девијација можеме да заклучиме дека вредностите се движат претежно околу 7 часа т.е. нема многу големи отстапки од оваа вредност. Минималната вредност на часови поминати во спиење е 1, а максималната вредност на оваа карактеристика изнесува 24. Дури 75% од испитаниците во просек спијат по 8 часа или помалку.

Овие мерења се прикажани и на табелата во прилог, на слика број 7. На слика 8, повторно може да се потврдат горенаведените заклучоци за варијација на податоците, особено за менталното и физичкото здравје.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **BMI** | 319795.0 | 28.325399 | 6.356100 | 12.02 | 24.03 | 27.34 | 31.42 | 94.85 |
| **PhysicalHealth** | 319795.0 | 3.371710 | 7.950850 | 0.00 | 0.00 | 0.00 | 2.00 | 30.00 |
| **MentalHealth** | 319795.0 | 3.898366 | 7.955235 | 0.00 | 0.00 | 0.00 | 3.00 | 30.00 |
| **SleepTime** | 319795.0 | 7.097075 | 1.436007 | 1.00 | 6.00 | 7.00 | 8.00 | 24.00 |

**Слика 7 „Одредени пресметани вредности за нумеричките карактеристики во податочното множество“**




**Слика 8 „Приказ на распределеноста на нумеричките податоци“**

## 4.2. Анализа на вредностите кои недостасуваат во сите карактеристики во податочното множество

За да добиеме точна слика за сите врски и релации помеѓу атрибутите во едно податочно множество и притоа да можеме да донесеме валидни заклучоци потребно е сите карактеристики т.е. атрибути кои ни се од интерес да немаат невалидни и несоодветни вредности, празни вредности или пак некои големи нелогични отстапувања. Конкретно во мојот случај, за да можеме да креираме валидни прашалници поврзани со сите фактори кои влијаат за добивање и развивање на срцево заболување потребно е претходно да го исфилтрираме податочното множество и да се справиме со несоодветните вредности.

Пред да направам анализа за да ги откријам несоодветните вредности во податочното множество, сметав дека ќе биде потребно да се справам со истите користејќи некоја од техниките за справување со вредностите кои недостасуваат, односно да отфрлам некои колони доколку биде преголем бројот на вредности кои недостасуваат, да отфрлам некои примероци кои што немаат поголем број на вредности, да ги пополнам вредностите што недостасуваат со средната вредност на податоците, со модата или пак со медијаната на соодветниот атрибут или пак да искористам некој од напредните техники за справување со вредности што недостасуваат како KNN класификација, MICE алгоритамот.

Но, согласно табелите дадена во продолжение, податочното множество во ниту еден атрибут нема вредности што недостасуваат, што значи дека нема потреба од преземање на никакви дополнителни акции за справување со вредности кои недостасуваат.

```
data.isnull().any()
```

| | |
|---|---|
| HeartDisease | False |
| BMI | False |
| Smoking | False |
| AlcoholDrinking | False |
| Stroke | False |
| PhysicalHealth | False |
| MentalHealth | False |
| DiffWalking | False |
| Sex | False |
| AgeCategory | False |
| Race | False |
| Diabetic | False |
| PhysicalActivity | False |
| GenHealth | False |
| SleepTime | False |
| Asthma | False |
| KidneyDisease | False |
| SkinCancer | False |

dtype: bool

```
data.isnull().sum()
```

| | |
|---|---|
| HeartDisease | 0 |
| BMI | 0 |
| Smoking | 0 |
| AlcoholDrinking | 0 |
| Stroke | 0 |
| PhysicalHealth | 0 |
| MentalHealth | 0 |
| DiffWalking | 0 |
| Sex | 0 |
| AgeCategory | 0 |
| Race | 0 |
| Diabetic | 0 |
| PhysicalActivity | 0 |
| GenHealth | 0 |
| SleepTime | 0 |
| Asthma | 0 |
| KidneyDisease | 0 |
| SkinCancer | 0 |

dtype: int64

**Слика 9 „Табеларен приказ на број на вредности кои недостасуваат по карактеристика во податочното множество“**

```
data.isnull().any()
```

| | |
|---|---|
| HeartDisease | False |
| BMI | False |
| Smoking | False |
| AlcoholDrinking | False |
| Stroke | False |
| PhysicalHealth | False |
| MentalHealth | False |
| DiffWalking | False |
| Sex | False |
| AgeCategory | False |
| Race | False |
| Diabetic | False |
| PhysicalActivity | False |
| GenHealth | False |
| SleepTime | False |
| Asthma | False |
| KidneyDisease | False |
| SkinCancer | False |

dtype: bool

**Слика 10 „Присутност на нул вредности во карактеристиките на податочното множество“**

Поради неприсутноста на невалидни и празни вредности во множеството од наш интерес, во овој чекор нема потреба од било каква филтрација на податоците и атрибутите и согласно ова може да преминеме на следниот чекор од ова истражување т.е. да ги

разгледаме и воочиме врските и релациите кои што постојат помеѓу сите карактеристики во податочното множество кое ги обработува клучните фактори за заболување на срцето.

## 4.3. Анализа на врските и поврзаноста помеѓу карактеристиките на податочното множество

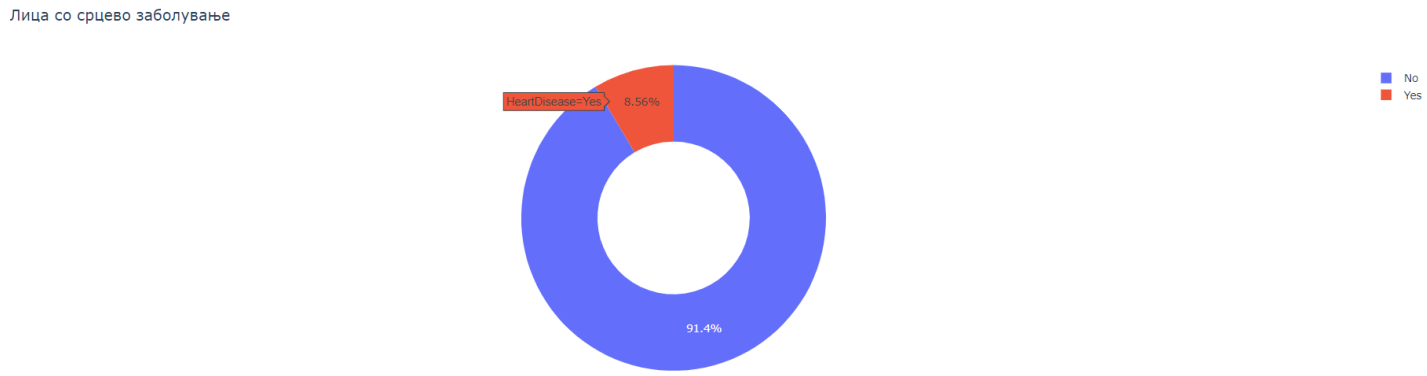
За да преминеме кон главниот дел на ова истражување, односно за да можеме да поставиме содржајни прашалници кои имаат смисла и притоа да ги истестираме перформансите на сите четири претставници на неструктурирани бази на податоци согласно различните критериуми, потребно е да навлеземе подлабоко во процесот на претпроцесирање на податоците и да ги испитаме различните врски и релации кои постојат меѓу сите 18 карактеристики од ова податочно множество.

Најважна карактеристика во ова податочно множество е карактеристиката која што содржи информации за постоењето на срцево заболување кај испитаниците. Од слика број 11, на која е прикажан процентот на лица со срцево заболување и процентот на лица без срцево заболување, а и од пита графиконот претставен на слика 12, јасно е видливо дека само 8.54% од испитаниците имале срцево заболување, а 91.4% немале срцево заболување.

```
print('Процент на лица кои имаат срцево заболување т.е срцеви проблеми: {:.4f}%'.format((len(data.HeartDisease[data.HeartDisease=='No']) / len(data)) * 100))
print('Процент на лица кои немаат срцево заболување т.е. срцеви проблеми: {:.4f}%'.format((len(data.HeartDisease[data.HeartDisease=='Yes']) / len(data)) * 100))
```

```
Процент на лица кои имаат срцево заболување т.е срцеви проблеми: 91.4405%
Процент на лица кои немаат срцево заболување т.е. срцеви проблеми: 8.5595%
```

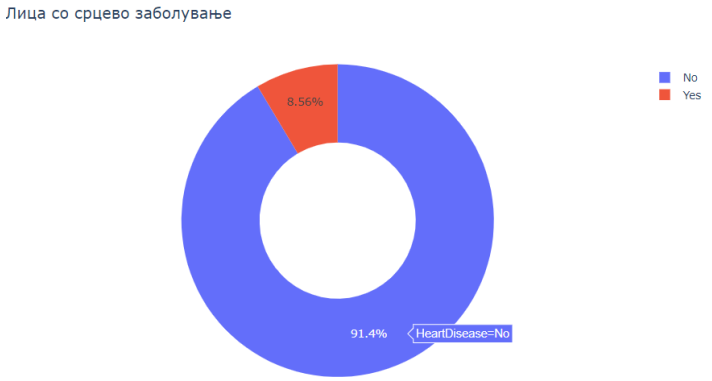
**Слика 11 „Процент на лица со/без срцево заболување“**




**Слика 12 „Пита графикон - Лица со срцево заболување“**

Во продолжение од главен интерес ќе ни бидат релациите помеѓу карактеристиката за срцево заболување и другите карактеристики.

Никотинот кој што го има во тутунот од цигарите предизвикува стеснување на артериите и зголемување на крвниот притисок. Активните пушачи имаат поголем ризик од заболување на срцето. Според пита графиконот на слика 13, дури 12.2% од лицата кои се активни пушачи се соочуваат со срцеви проблеми и некаков вид на срцево заболување. За разлика од активните пушачи, 6.03% од лицата кои не пушат цигари имаат срцево заболување, според пита графиконот на слика 14. Според овие два пита графикони, активните пушачи имаат двојно повеќе срцеви заболување во однос на лицата кои не пушат цигари.




**Слика 13 „Пита графикон - Колкав процент од луѓето кои се активни пушачи имаат срцево заболување“**


**Слика 14 „Пита графикон – Колкав процент од луѓето кои не пушат цигари имаат срцево заболување“**

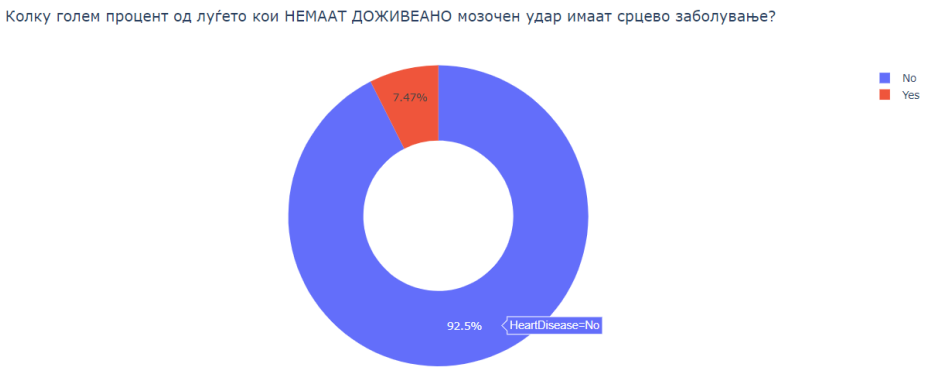
Од друга страна, алкохолот во повеќето случаи е штетен за нашиот организам, но конкретно, може да биде и превенција за срцеви заболувања. Алкохолот ја подобрува циркулацијата и го спречува згрутчувањето на крвта, а го зголемува добриот холестерол, па согласно пита графиконите на сликите 15 и 16, 5.24% од лицата кои консумираат алкохол имаат срцево заболување, а 8.8% од лицата кои не консумираат алкохол имаат срцево заболување. Бројот на лица кои не консумираат алкохол, а имаат срцево заболување е значително поголем.


**Слика 15 „Пита графикон - Колкав процент од испитаниците консумираат алкохол и имаат/немаат срцево заболување“**


**Слика 16 „Пита графикон - Колкав процент од испитаниците не консумираат алкохол и имаат/немаат срцево заболување“**

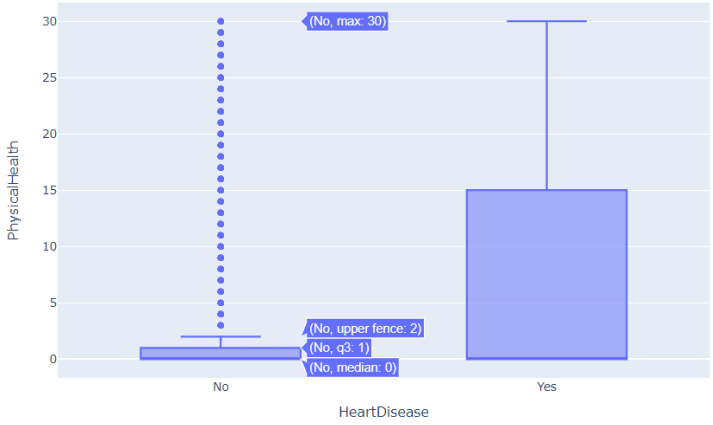
Голем број од предизвикувачите на срцеви заболување се предизвикувачи и на мозочен удар, па поради ова многу често луѓето кои имаат срцеви проблеми, имаат предиспозиции за добивање на мозочен удар и обратно. Според пита графиконите на сликите 17 и 18, значително поголем е бројот на лица кои имале мозочен удар и имаат срцево заболување т.е 36.4% од испитаниците имале и мозочен удар и срцево заболување, а 7.47% од испитаниците имале само срцево заболување.


**Слика 17 „Пита графикон - Колкав процент од лицата кои доживеале мозочен удар имаат/немаат срцеви проблеми“**



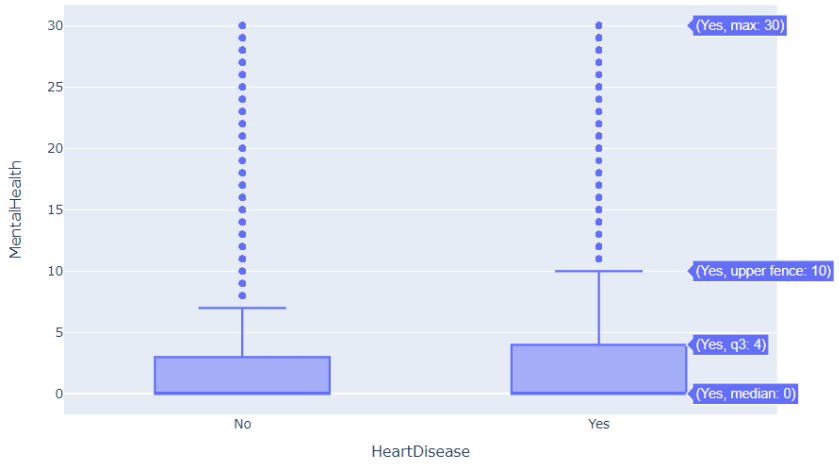

**Слика 18 „Пита графикон - Колкав процент од лицата кои немаат доживеано мозочен удар имаат/немаат срцеви проблеми“**

Под физичко здравје се мисли на целокупното здравје и форма на нашето тело т.е. дали во последните 30 дена, сме го „нарушиле“ истото , односно дали сме се стекнале со некоја физичка повреда и дали сме чувствувале болки итн. Од графичката репрезентација на слика 19, можеме да го воочиме соодносот на атрибутите Физичко здравје и Срцево заболување. Оваа репрезентација, која е поделена на квартали (25%, 50%, 75%) според бројот на испитаници, кои во случајот имаат или немаат срцево заболување и имале нарушено физичко здравје во опсег од 30 дена. Според овој приказ доколку сте имале нарушено физичко здравје во последните месец дена, можно е да добиете или веќе да имате срцево заболување.




**Слика 19 „Графичка репрезентација според квартали на соодносот на атрибутот Физичко Здравје и атрибутот Срцево заболување“**

Како со физичкото здравје, слична е и ситуацијата и со менталното здравје, која може да се увиди од графичкиот приказ на слика број 20. Поголеми се шансите да развиете срцево заболување доколку имате нарушено ментално здравје во поголемиот дел од деновите во месецот.




**Слика 20 „Графичка репрезентација според квартали на соодносот на атрибутот Ментално Здравје и атрибутот Срцево заболување“**

Секое нарушување на човековото здравје си носи свои последици, па сосема е очекуван резултатот кој што е добиен на двата пита графикони дадени во продолжение. Луѓето кои имаат срцево заболување, може да се соочат со одредени физички слабости како што се потешкотиите во движењето и качувањето скали. Токму од двата пита графикони можеме да увидиме дека 22.6% од испитаниците кои имале срцево заболување имаат и потешкотии во движењето, а само 6.3% од испитаниците кои имале срцево заболување се изјасниле дека немаат потешкотии во движењето.


**Слика 21 „Пита графикон - Колкав процент од испитаниците имаат потешкотии во движењето и притоа имаат/немаат срцево заболување“**


**Слика 22 „Пита графикон - Колкав процент од испитаниците немаат никакви потешкотии во движењето и притоа имаат/немаат срцево заболување“**

Според повеќето научни истражувања во периодот од 2020 година до пишување на ова истражување т.е. дипломска работа, мажите имаат поголеми шанси да развијат некое срцево заболување за разлика од спротивниот пол т.е. жените. Некои од причините за поголемиот број на мажи кои се соочуваат со срцево заболување се потешките физички работни места, помалата грижа за сопственото здравје и игнорирањето на симптомите, поголемиот крвен притисок, холестеролот, дијабетесот итн. Дека поголем процент од лицата од машки пол имаат срцеви заболување наспроти лицата од женски пол сведочи и податочното множество од наш интерес, согласно пита графиконите на сликите 23 и 24, 10.6% од мажите имаат срцево заболување наспроти 6.69% на лица од женски пол кои имаат срцево заболување.

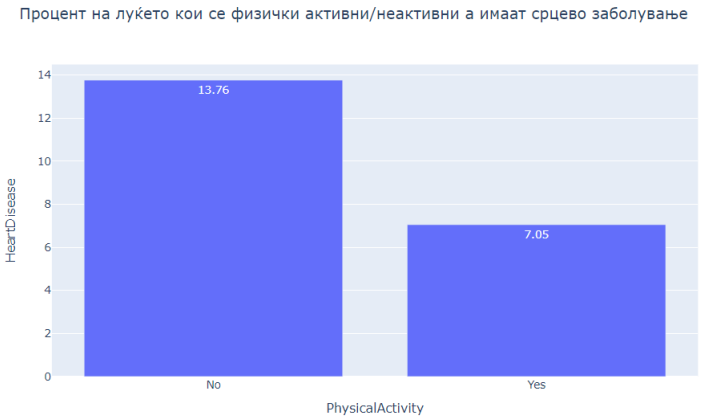

**Слика 23 „Пита графикон - Колкав процент од жените имаат/немаат срцево заболување“**


**Слика 24 „Пита графикон - Колкав процент од мажите имаат/немаат срцево заболување“**

Живеењето во 21 век носи многу поволности, но и брз начин на живот. Општопознато е дека како што старее човечкиот организам така се посклон е на разни заболувања вклучително и развивањето на срцеви заболувања. Според графиконот прикажан во продолжение, интересен и застрашувачки е фактот дека постојат млади индивидуи со срцеви проблеми. Сепак кривата расте на десно, т.е. постарите испитаници имаат поголем процент на срцеви заболувања во однос на помладите.

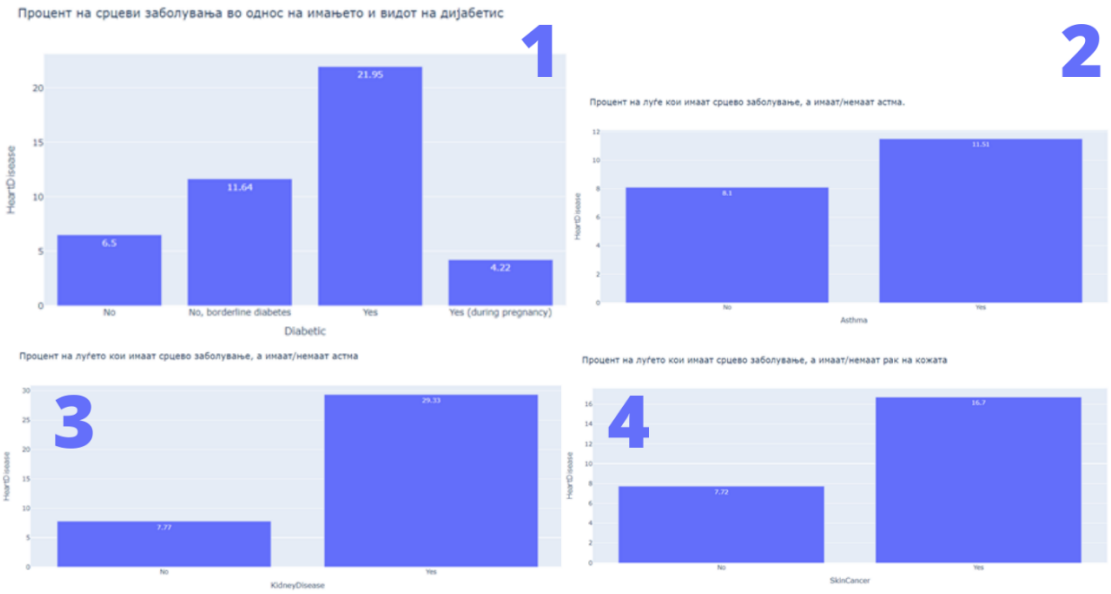

**Слика 25 „Графички приказ на процентот на лица со срцеви заболување според старосната група“**

Физичката активност го зајакнува нашето срце и го намалува ризикот за појавување на срцево заболување намалувајќи го крвниот притисок, подобрувајќи го управувањето со нивото на холестерол, инсулин и други масти во крвта, како и намалувајќи го нивото на Ц реактивен протеин во нашето тело т.е. знакот за воспаление. Овие факти се клинички докажани, но со ова истражување можеме и да ги потврдиме, водејќи се според графиконот во продолжение. Дури 13.76% од испитаниците кои се изјасниле дека се физички неактивни имаат срцево заболување, додека пак бројот на лица кои се физички активни и имаат срцево заболување е речиси двојно помал т.е. 7.05%.



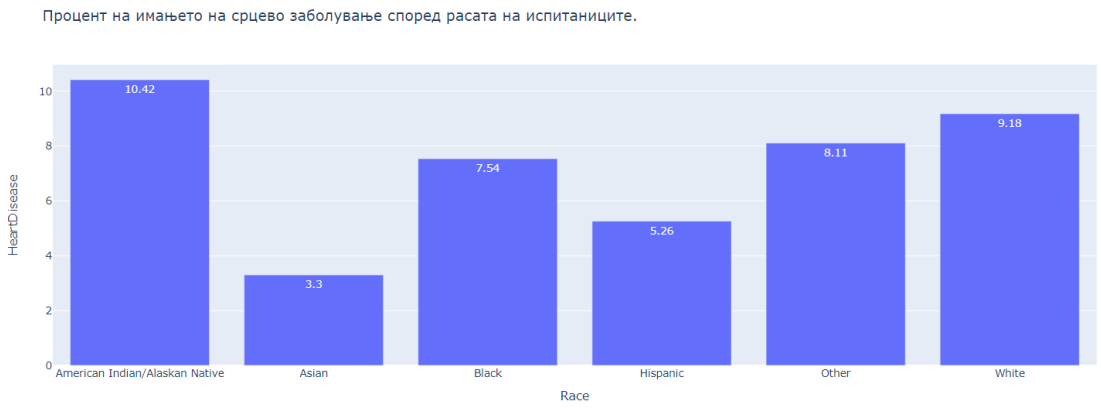

**Слика 26 „Графички приказ на соодносот на процентот на лица кои се физички активни/неактивни и имаат срцево заболување“**

Имунитетот е значително намален и нашиот организам е исцрпен кога нашето тело се бори со бактерии, вируси и разни болести, особено кога станува збор за рак на кожата, проблеми со белите дробови како што е астма, проблеми со шеќерот во крвта како што е дијабетес и проблеми со бубрезите како песок, камења итн. Од графиконите прикажани на слика 27, можеме да увидиме дека поголеми шанси за развивање на срцево заболување имаат лицата со дијабетес и тоа 21.95% и 4.22% дијабетес развиен во бременост (слика 27, графикон број 1), лицата со астма 11.53% (слика 27, графикон број 2), лицата со заболување на бубрезите 29.33% (слика 27, графикон број 3) и лицата со рак на кожата 16.7% (слика 27, графикон број 4).




**Слика 27 „Графички приказ бројот на испитаници кои имаат срцево заболување а имаат/немаат 1.Дијабетес, 2. астма, 3. Заболување на бубрезите, 4. Рак на кожата “**

Расата како карактеристика во ова податочно множество е за генерална класификација на испитаниците поради анонимноста на прашалниците со чија помош е создадено ова податочно множество. Сепак, интересно е да се увиди дека најмногу испитаници имаме од бела раса, а сепак според графиконот во продолжение истите не се со најголем број на срцеви заболувања. Испитаниците кои се класифицирале како американци со индиско потекло имаат најголем процент на лица кои имаат срцеви проблеми, дури 10.42%, а истите се најмалку бројни во ова податочно множество.




**Слика 28 „Графички приказ на лица со срцево заболување според раса“**

Според податоците прикажани на графиконот во продолжение, лицата кои просечно поминуваат над 18 часови во спиење, дневно, сочинуваат значително поголем процент на лица со срцеви заболувања т.е над 20.59%. Од друга страна, препорачливо време на проспиени часови дневно е во опсегот околу 7 до 9 часа, според графиконот најмал број на испитаници кои имале срцево заболување, спиеле по 7 часа и нивниот број изнесува 6.47%.


**Слика 29 „Графички приказ на лицата со срцево заболување според проспиени часови дневно“**

Како што спомнав претходно, грижата за сопственото здравје е многу важна и доколку ја запоставуваме многу брзо може да ни се одрази на истото. На генералното здравје влијаат и лошиот сон, физичката неактивност, пушењето цигари, постоењето на некои заболувања, нарушеното ментално здравје итн. Односно можеме да кажеме дека на генералното здравје влијаат сите досега обработени карактеристики. Според графиконот во прилог, најголем процент на срцеви заболувања, 34.1%, има групата на испитаници кои го оцениле своето генерално здравје како лошо.


**Слика 30 „Графички приказ на процентот на испитаници кои имаат срцево заболување според генералната оценка на сопственото здравје“**

Индексот на телесна маса е еден од најважните атрибути во ова податочно множество. Поради неговата важност и поврзаност со многу други атрибути, го обработуваме откако ги разгледавме и обработивме сите други атрибути и нивната поврзаност со атрибутот за присутност на срцево заболување кај испитаниците.

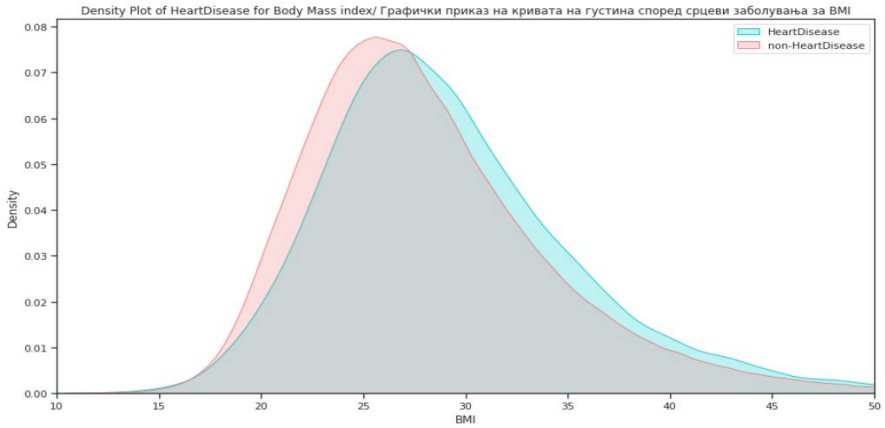
Според светската здравствена организација прекумерната тежина е еден од клучните фактори за добивање на срцев удар или за појава на некое од срцевите заболувања. Пред да го добијам дијаграмот за поврзаноста меѓу индексот на телесна маса и појавата на срцеви заболувања очекував дека ќе добијам резултати во кои тврдењето на СЗО ќе биде јасно видливо, но во мојот случај не може да се каже дека лицата кои имаат индекс на телесна маса над 24.9, број што претставува горна граница за здрава телесна маса, имаат евидентно поголем број на срцеви заболувања во однос на испитаниците кои немале срцево заболување. Но по направената детална анализа на атрибутот индекс на телесната маса, прикажана на слика 31, можеме да увидиме дека дури 99% од испитаниците имаат индекс на телесна маса под 48. Иако поголемиот број од испитаниците немаат срцево заболување, според дијаграмот најголем број на лица кои што имаат проблеми со срцето имаат индекс на телесна маса во рангот од 10 до 50, што вклучува лица со индекс на телесна маса под, во границите и над вредноста која се смета за здрав индекс на телесна маса.


**Слика 31 „Дијаграм на поврзаност на карактеристиките Индекс на телесна маса и Срцево заболување“**

```
count    319795.000000
mean         28.325399
std           6.356100
min          12.020000
25%          24.030000
50%          27.340000
70%          30.410000
80%          32.690000
95%          40.180000
97%          42.910000
99%          48.660000
max          94.850000
Name: BMI, dtype: float64
```

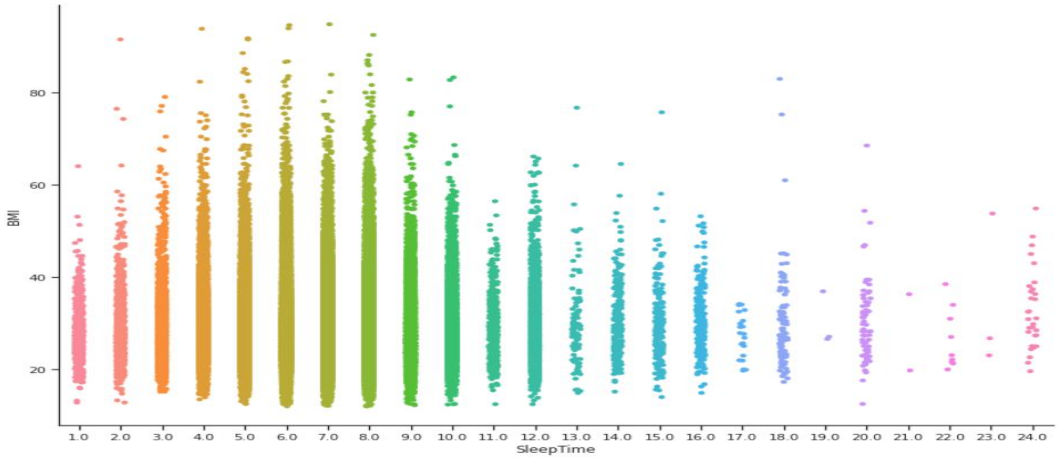
**Слика 32 „Детална анализа на податоците од карактеристиката за Индекс на телесна маса“**

Интересно е да се разгледа слика 33, графичкиот приказ на кривата на густина на карактеристиката индекс на телесна маса според постоењето на срцево заболување. Според овој графички приказ најголем број на од испитаниците кои имале срцево заболување имаат индекс на телесна маса околу 26, што не е голема отстапка од горната граница на здрава телесна маса. Двете криви на густина прикажани на графичкиот приказ имаат нормална Гаусова распределба, минимално поместена на десно.



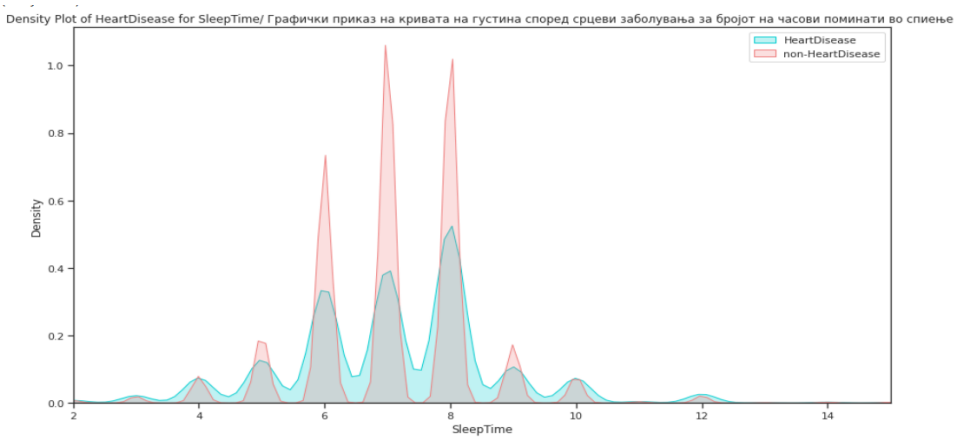

**Слика 33 „Графички приказ на кривата на густина на податоците на карактеристиката Индекс на телесна маса според карактеристиката за срцеви заболувања“**

Според дијаграмот на поврзаност на карактеристиките индекс на телесна маса и време поминато во спиење даден во продолжение, можеме да заклучиме дека испитаниците кои што спијат просечно во опсег од 7 до 9 часа т.е. спијат препорачлив број на часови дневно, немаат некоја специфична тежина. Добриот сон е важен за да не се појават проблеми со прекумерна тежина, од дијаграмот на поврзаност можеме да заклучиме дека поголемиот број од лицата кои што просечно спијат 17,18,20 и 22 часа, што не е препорачливо време за спиење, имаат здрава телесна маса т.е. нивниот индекс на телесна маса е во опсегот од 18.5 до 24.5. Но, да не заборавиме дека овие четири категории имаа голем број на испитаници кои имаат заболување на срцето.



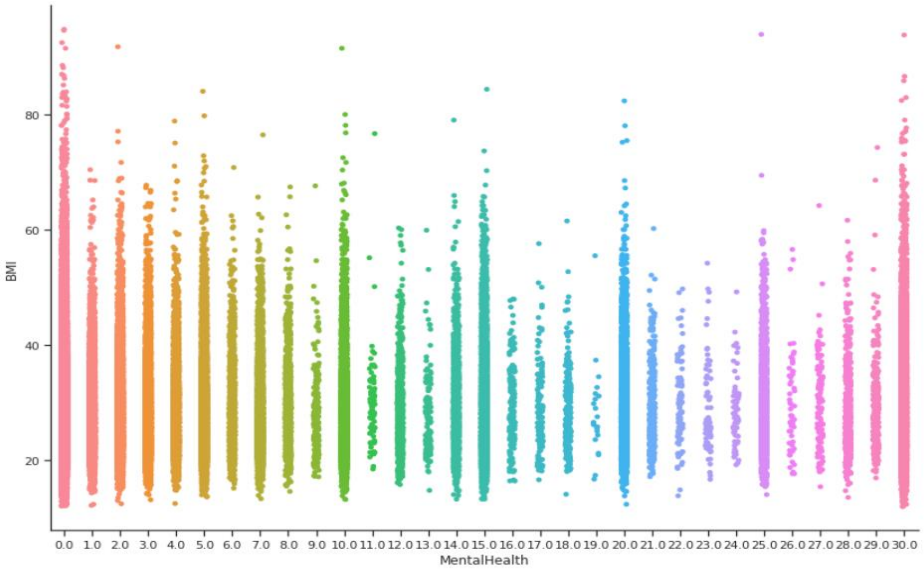

**Слика 34 „Дијаграм на поврзаност на карактеристиките Индекс на телесна маса и Време поминато во спиење“**

За да направиме споредба на индексот на телесна маса, просечното време поминато во спиење и присуството на срцево заболување, во овој дел го додадов и графичкиот приказ на кривите на густина на податоците од карактеристиката просечно време поминато во спиење според постоењето на срцево заболување. Од графичкиот приказ на слика 35, можеме да увидиме дека најголем број на испитаници кои имаат срцево заболување спиеле препорачлив број на часови т.е. 8 часа. А, доколку се навратиме на дијаграмот на поврзаност прикажан на слика 33, можеме да увидиме дека испитаници кои спиеле 8 часа имаат индекс на телесна маса претежно над границата на нормална телесна маса. Ова ни кажува дека иако спиеме препорачлив број, ако имаме зголемена телесна тежина може да развиеме срцево заболување.




**Слика 35 „Графички приказ на кривата на густина на податоците на карактеристиката часови поминати во спиење според постоењето на срцево заболување“**

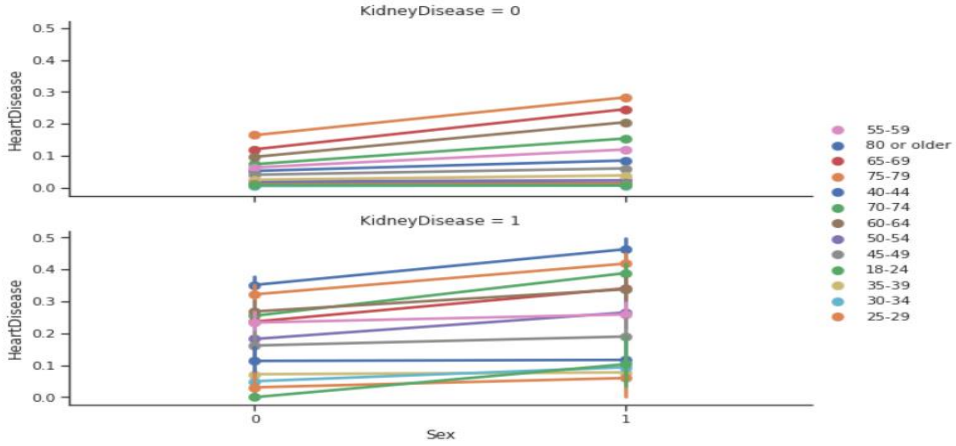
Соочувањето со проблем со тежина, над и под нормалната граница, може да доведе до појава на ментални растројства и нарушување на менталното здравје. Но, овој проблем не е еднонасочна улица т.е. и нарушувањето на менталното здравје може значително да влијае на тежината. Справувањето со менталните нарушувања е доста тежок и макотрпен процес и не е здраво и нормално индивидуата да има нарушено ментално здравје над 15 дена во месецот, токму затоа нашиот интерес е насочен во делот од дијаграмот во прилог во кој се разгледуваат испитаниците кои имале нарушувања на менталното здравје во поголемиот дел од месецот. Можеме да увидиме дека имаме испитаници кои иако се соочиле со ментално нарушување во опсег од 15 до 30 дена, сепак сеуште се во здрава граница на индексот на телесна маса т.е. во опсег од 18.5 до 24, но се значително помал број во однос на лицата кои страдале дури 30 дена од месецот од ментално нарушување, кои имаат прекумерна тежина. Има и испитаници кои имаат индекс на телесна маса под нормалната граница. Според дијаграмот, доста голем број од испитаниците имале тежина над и под границата на здрава телесна маса и нарушување во менталното здравје во поголемиот дел од месецот. Но, постојат и лица кои воопшто немале нарушување во менталното здравје, а имаат тежина во, над и под границите на нормалата.




**Слика 36 „Дијаграм на поврзаноста помеѓу карактеристиките Индекс на телесна маса и Ментално здравје“**

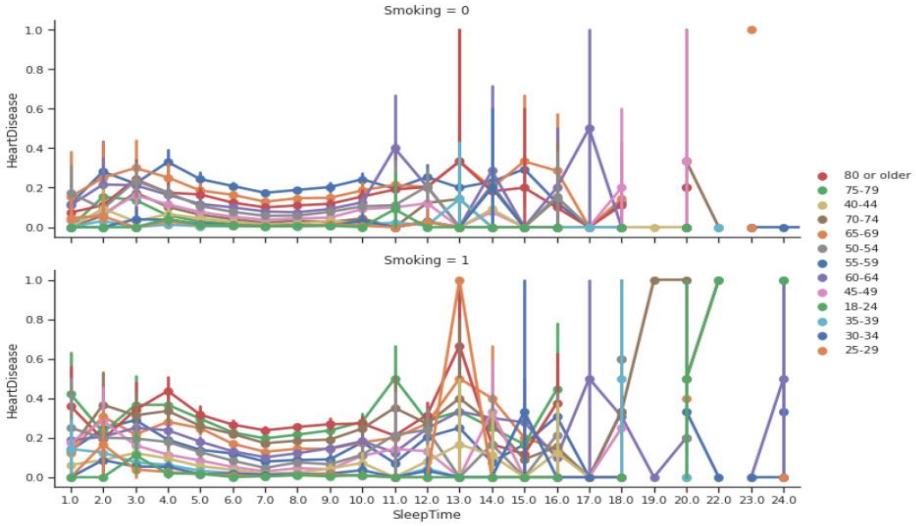
Досега испитуваме поврзаност помеѓу само две карактеристики, ставајќи ја во фокусот карактеристиката за постоењето на срцево заболување. Во следниот чекор, ќе обработиме два графички прикази кои ја прикажуваат застапеноста на срцево заболување меѓу испитаниците според три други карактеристики.

Според графичкиот приказ на слика 37, можеме да увидиме дека мажите кои се на возраст од 80 години или повеќе и имаат заболување на бубрезите претставуваат и најголем број на лица кои се соочуваат со срцево заболување. За нијанса помалку бројни се жените кои имаат срцево заболување, а се на иста возраст како и мажите и притоа имаат заболување на бубрезите.



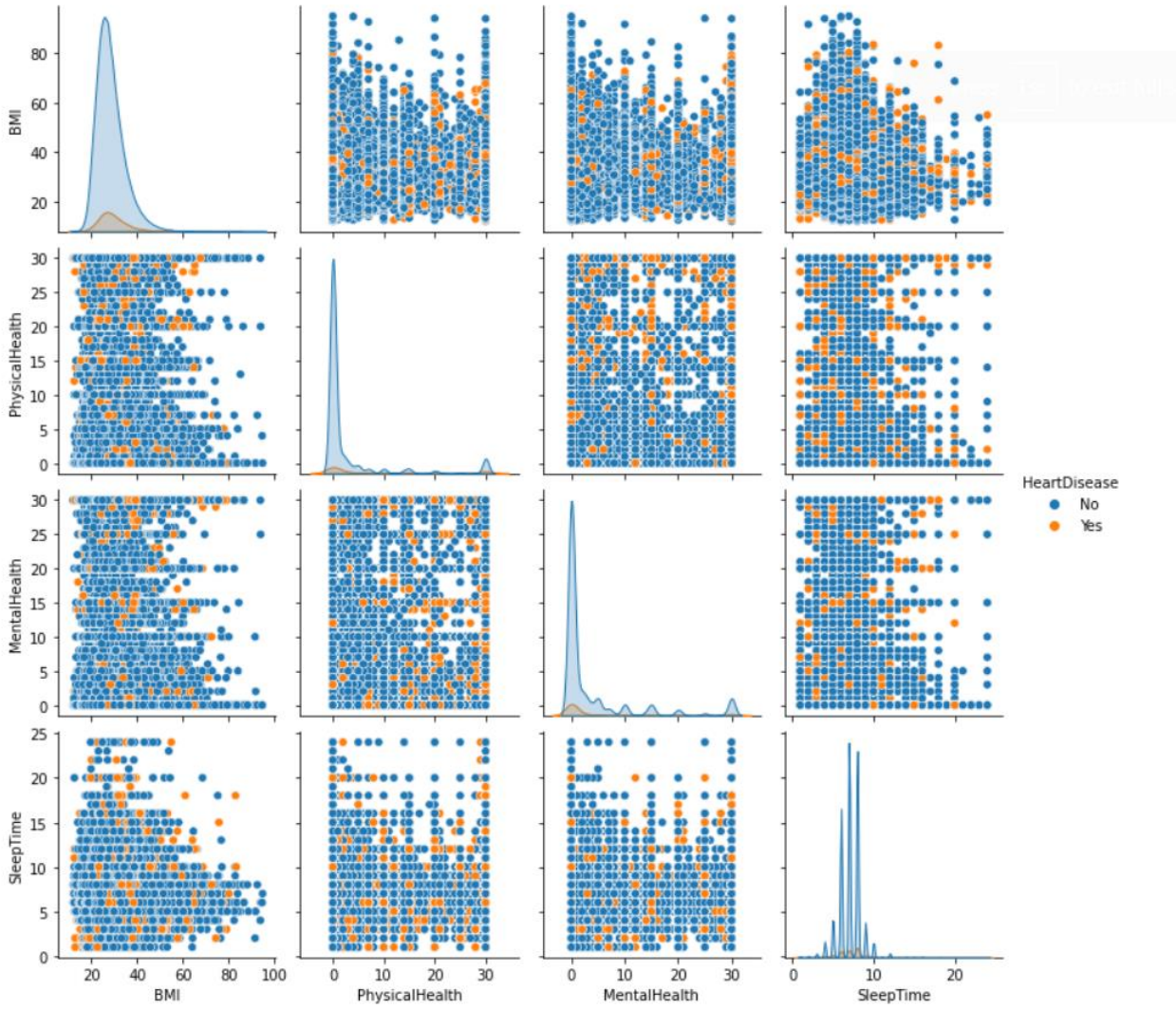

**Слика 37 „Графички приказ на застапеноста на Срцево заболување според Полот, Старосната Група и Заболувањето на бубрегот“**

Од графичкиот приказ на застапеноста на срцево заболување според бројот на проспиени часови, старосната група и активното пушење цигари, слика 38, евидентно е дека најголем број на испитаници кои имале срцево заболување се на возраст од 65 до 69 години и притоа спиеле 13 часа и активно пушеле цигари. Голем е и бројот на испитаници на возраст од 70 до 74 години кои спиеле 19 часа и притоа активно пушат цигари и се соочуваат со срцево заболување.




**Слика 38 „Графички приказ на застапеноста на Срцево заболување според бројот на проспиени часови, старосната група и активното пушење цигари“**

За да може да ги разгледаме корелациите помеѓу сите атрибути на податочното множество, сите категориски податоци ги претворив во нумерички, бидејќи имаат само две или три уникатни вредности. Во прилог е даден визуелен приказ на матрицата на корелација на нумеричките вредности. Во визуелниот приказ на секоја од корелациите е додаден и атрибутот за срцеви заболување т.е. секој графички приказ претставува корелација меѓу два атрибути и дополнително се разгледува и постоењето на срцево заболување. Во овој графички приказ, а и од матрицата на корелација дадена на слика 40, можеме да забележиме дека немаме некоја забележителна позитивна корелација меѓу нумеричките податоци.




**Слика 39 „Графички приказ на матрицата на корелација“**

Според самата матрица на корелација дадена во продолжение можеме да увидиме дека постои позитивна корелација, од 0.43, помеѓу атрибутите физичко здравје и проблеми со движењето т.е. дека луѓето кои имале некоја повреда во изминатиот месец се соочуваат со проблеми во движењето.

Слаба позитивна корелација од 0.29 имаме меѓу атрибутите Ментално здравје и Физичко здравје т.е. доколку имаме некоја повреда, можно е и да доживееме ментално растројство.

Интересно е однесувањето на атрибутот дијабетес т.е. истиот има слаба позитивна корелација со атрибутите:

- проблеми во движењето, од 0.21, односно лицата кои имаат дијабетес може да се соочуваат и со проблеми во движењето,
- индекс на телесна маса, од 0.20, т.е. лицата кои имаат дијабетес многу често имаат проблеми и со прекумерна или премала телесната маса,
- срцево заболување, од 0.17, т.е. лицата со дијабетес може да развијат и срцеви проблеми

а дијабетес има негативна корелации со генералното здравје, од -0.25, односно доколку проблемите со дијабетес се намалуваат, генералното здравје се подобрува.

Атрибутот за срцево заболување е позитивно корелиран со:

- Физичкото здравје, 0.17, односно доколку имаме срцеви проблеми можно е да имаме нарушено физичко здравје т.е. лимитирани физички можности,
- Проблеми во движењето, 0.20, т.е. лицата со срцеви проблеми се соочуваат и со проблеми во движењето,
- Мозочен удар, 0.20, односно соочувањето со срцеви проблеми може да биде и предизвикувач на мозочен удар и обратно,
- Проблеми со бубрезите од 0.15, т.е. проблемите со бубрезите може да се појават кај лица со срцево заболување или пак во обратната насока,
- Пушење цигари, 0.11, како што беше и спомнато во анализата на врските меѓу овие два атрибути, активните пушачи може да се соочат со срцеви заболувања,

Истиот има негативна корелација со :

- Генералното здравје, -0.23, доколку лицето има проблеми со срцето, генералното здравје е намалено, па при подобрување на клиничката слика на срцето, генералното здравје се подобрува,
- Физичката активност, од -0.10, т.е. зголемената физичката активност може да ги намали проблемите со срцето поради зголемување на циркулацијата, намалување на шеќерите во крвта, холестеролот итн.,
- Консумирањето алкохол, -0.03 , иако на прв поглед изгледа чудно, консумирањето на алкохол во умерени количини може да го подобри здравјето на срцето т.е. да ги намали проблемите со срцето.

Овие корелации се воочливи во матрицата на корелација, но и на слика 42, на која е прикажана јачината на корелациите на атрибутот срцево заболување со другите атрибути, како и на слика 43 на која јасно визуелно се дадени јачините на позитивните и негативните корелации на овој атрибут.

Можеме да увидиме дека воопшто немаме никаква корелација помеѓу атрибутите пол и дијабетес т.е. дали лицето има проблеми со дијабетес воопшто не зависи од полот. Но,

полот има слаба позитивна корелација со атрибутот срцево заболување т.е. како што увидовме во горе направената анализа мажите се посклони на срцеви заболувања поради потешките физички активности, помалата грижа за сопственото здравје итн. Полот не е корелиран ни со просечното време на спиење, односно бројот на просечно проспиени часови зависи од индивидуа, до индивидуа, а не од полот. И физичката активност не е корелирана со полот. Дали физички ќе бидеме активни е наш избор и апсолутно овој факт е непроменлив зависно нашиот пол.

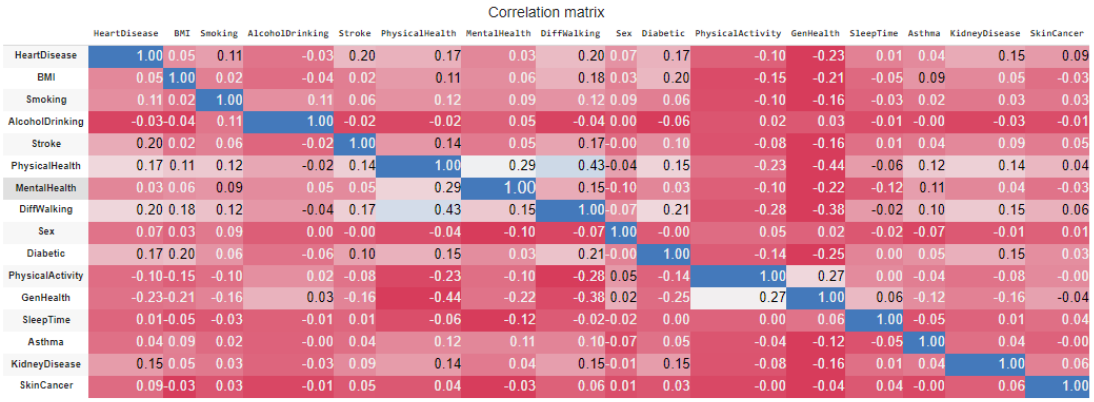
Никаква корелација не постои и помеѓу атрибутите рак на кожата и физичка активност т.е. ракот на кожата не ги спречува лицата да бидат физички активни.

При детално разгледување на негативните корелации прикажани на матрицата на корелација, можеме да увидиме дека најголем број на вакви корелации имаат атрибутите консумирање на алкохол и генералното здравје.

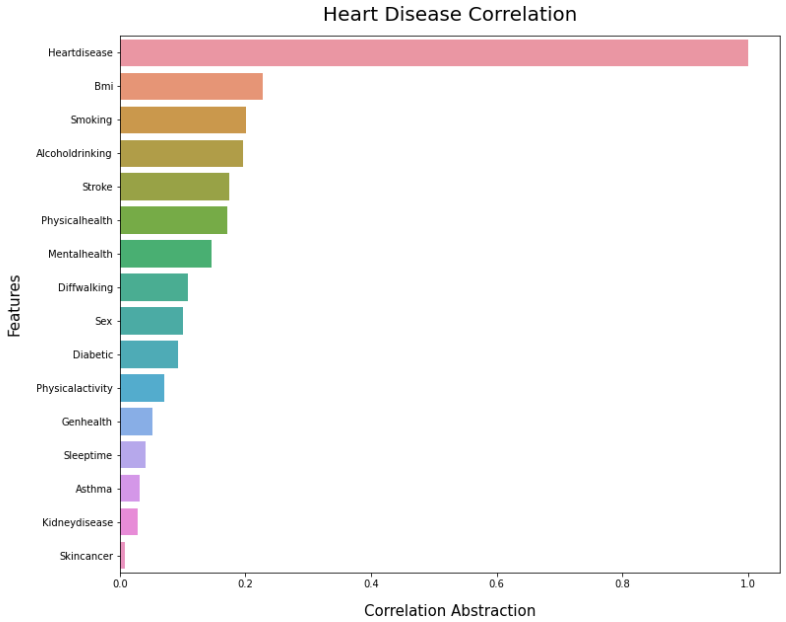
Умерената консумација на алкохол, за нијанса го намалува ризикот од добивање на мозочен удар, ги намалува проблемите со индексот на телесна маса, физичкото здравје, движењето, дијабетесот, бубрезите.

Генералното здравје покрај негативната корелација со заболувањата на срцето, е во негативна корелација и со индексот на телесна маса, пушењето цигари, мозочен удар, физичкото здравје, менталното здравје, проблемите во движењето, дијабетесот, астма, заболување на бубрези и рак на кожата. Од горенаведеното можеме да заклучиме дека секое штетно влијание и негативна промена во организмот се одразува на генералното здравје. Како што вели Б.К.С. Ајенгар „Здравјето е состојба на целосна хармонија на телото, умот и духот“.

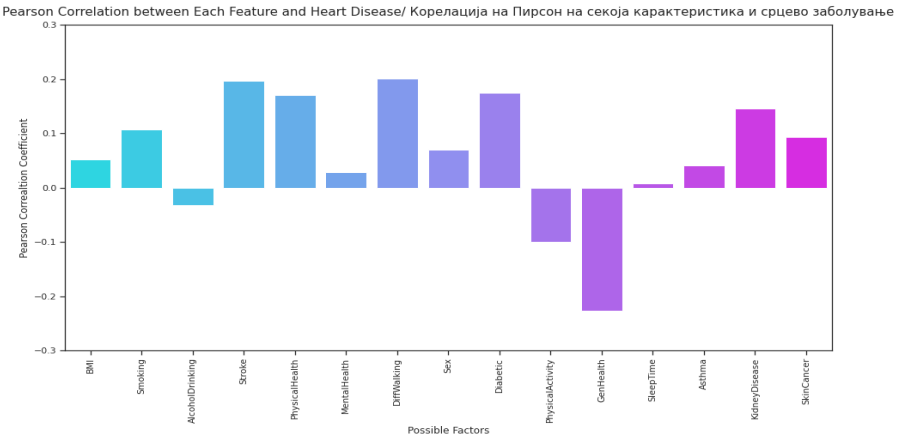
Од матрицата на корелација можеме да ги увидиме сите меѓусебни поврзаности на податоците, но за ова истражување главен фокус е ставен на заболувањето на срцето и на поголемите вредности во матрицата на корелација.




**Слика 40 „Матрица на корелација“**




**Слика 41 „Матрица на корелација на атрибутот срцево заболување со другите атрибути“**




**Слика 42 „Корелација на Пирсон на атрибутот срцево заболување со другите атрибути“**

До сега се стекнавме со знаења за обликот, врските и тенденциите на атрибутите од податочното множество кое во фокусот ги става клучните 17 фактори за добивање на некој вид на срцево заболување. Овој дел ни претставува солидна подлога за пишување на прашалници кои ќе имаат смисла и кои ќе ги поткрепат информациите добиени во овој дел. Како што е спомнато на неколку наврати во досегашниот тек на истражувањето, прашалниците се многу важен дел од оваа дипломска работа бидејќи со нивна помош ќе ги анализираме перформансите, синтаксата и интуитивноста на четири претставници од различни типови на неструктурирани бази на податоци.

# 5. Импортирање и поврзување на четирите неструктурирани бази на податоци соодветно, Amazon DynamoDB, Apache Cassandra, MongoDB, Neo4j

**Amazon DynamoDB** е претставник на т.н. key-value pair вид на неструктурирани бази на податоци. Оваа база на податоци нуди вградена безбедност, континуирана поддршка, автоматизирана репликација на повеќе региони, кеширање во меморија и многу алатки за импортирање и експортирање на податоци. Базата е дел од многуте алатки кои ги нуди пакетот Амазон Веб Сервис т.е. AWS. Можности за внесување на податоците кои ги нуди оваа база на податоци се преку т.н. lambda функции и s3 кошници, но истите функционираат само за мал број на податоци, па поради оваа причина, податочното множество го внесов во базата со помош на програмскиот јазик Python, користејќи ја библиотеката boto3. Најпрво секој ред од .CSV датотеката го претворив во .JSON датотека и во базата го внесов со помош на клиент на ниско ниво што го претставува Amazon DynamoDB, од библиотеката boto3. За да може успешно да се внесат податоците потребно беше да се изгенериират и внесат два вида на клучеви и тоа клуч за пристап до базата во т.н. Амазон Веб сервис и таен клуч. По создавање на нова табела и внесување на податоците во истата, ги користев двата јазици за пребарување компатибилени со SQL за Amazon DynamoDB, PartiQL и HiveQL. За имплементирање на прашања со PartiQL, користев DynamoDB API кое претставува специјализирана алатка за поставување на прашалници со помош на PartiQL. Некои покомплексни прашања имаа потреба од извишување со помош на јазикот HiveQL во апликацијата Hive која во позадина работи на Hadoop. Hadoop е збирка софтверски алатки со отворен код за решавање на проблеми кои вклучуваат огромни количини на податоци и пресметување. Hive претставува алатка која користи MapReduce се со цел на едноставен начин, со едноставно поставено прашање со помош на HiveQL да врати одговор, без притоа ние да го пишуваме MapReduce делот.

**Apache Cassandra** е претставник на вториот вид на неструктурирани бази на податоци т.н. Column-oriented. Архитектурата на оваа база на податоци е дело на тим од компанијата Facebook, се со цел да претставува спој помеѓу Amazon DynamoDB и BigTable на компанијата Google. Apache Cassandra го зема најдоброто од другите две неструктурирани бази на податоци и е создадена се со цел да ги задоволи барањата за системи кои зафаќаат големи размери. Оваа база на податоци е позната како „Ламборџини“ во светот на неструктурираните бази на податоци и е користена од многу компании кои се дел од т.н. Fortune 100, годишна листа на американското списанието Fortune во која се претставени 100 најголеми компании според приходите. Во оваа база на

податоци на едноставен начин го внесов податочното множество, со помош на едноставна команда за внес на .CSV датотека. За извршување на прашалниците во следниот дел од ова истражување користев локална верзија на Apache Cassandra и работев преку cqlsh алатката која претставува интерфејс на командната линија за интеракција со Apache Cassandra со користење на т.н. CQL (Cassandra Query Language).

Иако Apache Cassandra е „Ламборџини“ во светот на неструктурираните бази на податоци, сепак таа не е најпопуларната неструктурирана база. Првото место на листата, со години наназад е зачувано за **MongoDB,** претставник на т.н. Document-oriented неструктурирани бази на податоци. Популарноста на оваа база на податоци се должи на специфичниот т.н. Документ модел на податоци, динамичката шема и експресивниот и едноставен јазик за пребарување. MongoDB нуди локална верзија, но и две дополнителни значајни алатки MongoDB Atlas и MongoDB Compass. MongoDB Atlas е целосно менаџирана база на податоци базирана на облак т.е. cloud based database, која што нуди можност за избор на провајдер на услуги во облак според нашите преференции, но меѓу другото, за нас најзначајно во овој случај е можноста за пристап до онлајн базата од апликации од: трета страна, други алатки на MongoDB како MongoDB Compass, локалната верзија на базата на нашиот компјутер. MongoDB Atlas ни го поедноставува значително користењето на оваа база и ни нуди можност на едноставен начин да создаваме и управуваме апликации од голем обем и да имаме пристап од било каде до потребните информации т.е. податоци. Од друга страна, MongoDB Compass е интерактивна алатка која има едноставен интерфејс за преглед, анализа, пребарување и оптимизирање на податоците во документ базата MongoDB. За нас, MongoDB Compass е значаен поради можноста за лесно внесување на податоци од .CSV датотека во MongoDB Atlas. Оваа алатка нуди едноставен графички интерфејс за прикачување на датотеки, како .CSV датотека, така и JSON датотека и нуди можност за избор на податочните типови на атрибутите во .CSV датотеката. Оваа алатка значително го олеснува процесот за внес на големи податочни множества, како што е и множеството во мојот случај и значително ни заштедува време. По внесот на податоците во базата, се поврзав со локалната верзија на MongoDB, користејќи ја алатката MongoDB Shell и по синхронизација со MongoDB Atlas, започнав со извршување на прашалниците.

Доколку немате никакви познавања од информациски технологии и бази на податоци, кога зборувате за складирање на податоци, најпрво, верувам дека би си ги замислиле податоците како јазли и би ги поврзале меѓусебно за да ги означите врските кои ги имаат. Но, кога навлезете во водите кои ги обработуваат сите видови на бази на податоци, особено релационите бази на податоци, таа слика која што сте ја имале се нарушува. За да ги увидите „врските“ меѓу податоците потребно е да ги „споите“ истите, наоѓајќи заеднички идентификатор т.е. клуч. Но, неструктурираните бази на податоци т.н. Graph-based databases ни нудат конечно да можеме да си ги зачуваме податоците во граф

со јазли и врски каков што си го замислуваме. Овие бази на податоци имаат револуционерна архитектура, нудејќи ни можност да пребаруваме низ релациите кои што постојат меѓу податоците т.е. јазлите, да додаваме параметри како на јазлите, така и на релациите и сликовито да можеме да ги увидиме сите информации кои ги пребаруваме и сите податоци кои ги имаме во базата. **Neo4j** е претставник на граф базите на податоци, користен од голем број на развивачи на софтвери, научници и компании поради своите извонредни перформанси, скалабилна аналитика, интелигентен развој на апликации и поддршка за развој на апликации со машинско учење и вештачка интелигенција. Оваа неструктурирана база на податоци овозможува лесен интерфејс за работа, нудејќи ни можност за креирање на податоци во неа и за пребарување низ истите со помош на т.н. јазик за пребарување Cypher. Cypher е сличен на SQL, но специјализиран за пребарување низ графови. Оваа карактеристика го прави многу поедноставен и по интуитивен за употреба, бидејќи за разлика од комплицираните join структури кои ги овозможува SQL за пребарување низ повеќе табели, во граф базираните структури табелите не постојат. Cypher е лесен за користење, лесен за учење и ги следи врските, во која било насока, за да открие претходно непознати врски и кластери. За внес на .CSV датотека во оваа база на податоци, ми беше потребна една команда напишана со помош на Cypher, со која ја внесов датотеката, ги означив имињата на атрибутите и нивните податочни типови и секој ред го внесов како посебен јазол. Поради природата на моето податочно множество немав можност да создадам врски меѓу податоците и целосно да ги увидам можностите кои ги нуди оваа моќна база на податоци. По создавање на графот, почнав со пишување на потребните прашалници кои ќе ги разгледаме во делот кој што следува во продолжение на оваа дипломска работа.

## 5.1. Модели на агрегација и имплементација

За да се запознаеме подобро со сите можности кои ги нудат горенаведените четири претставници на неструктурираните бази на податоци, и притоа да ги тестираме нивните перформанси, потребно е да напишеме неколку прашања користејќи ги различните модели на агрегација. За да имаат поставените прашања смисла и правило да ги обработуваат поврзаностите на податоците, во ова истражување, како прв чекор ми беше да направам детална анализа на податочното множество. Во овој чекор, ќе тестираме петнаесет прашања добиени според веќе направената анализа кои повторно ќе ги обработат и потврдат поставените хипотези за клучните фактори кои влијаат за добивање на срцево заболување и водно кои се индиректни предизвикатели за другите болести, како астма, заболување на бубрезите, мозочен удар, рак на кожата.

За детално да ги испитам перформансите кои ги нудат четирите бази на податоци, соодветно Amazon DynamoDB, Apache Cassandra, MongoDB, Neo4j, ги користев следните модели на агрегација:

- Минимална вредност на соодветен атрибут од податочното множество т.е. Minimum(MIN)
- Максимална вредност на соодветен атрибут од податочното множество т.е. Maximum (MAX)
- Просечна/Средна вредност на атрибутите од наш интерес од податочното множество т.е. Average (AVG)
- Сума на вредности од податочното множество т.е. SUM
- Број на записи во податочното множество т.е. COUNT
- Групирање на записи според даден атрибут/атрибути т.е. GROUP BY
- Map-Reduce: Функција составена од два дела т.е. две подфункции, Map дел кој ни служи за групирање на податоци од податочното множество според соодветен услов т.е. во случајот клуч-вредност пар и Reduce дел кој претставува засебна функција во која се филтрираат добиените податоци и се селектираат само податоците кои задоволуваат одредено барање.

Специфично за Map-Reduce функцијата на агрегација е важно да се нагласи дека повеќето од прашалниците кај MongoDB, Apache Cassandra, Neo4j можат да се изведат без користење на оваа функција бидејќи се смета за застарена. Конкретно Neo4j не го подржуваа воопшто овој концепт. Но, сепак на пример кај Amazon DynamoDB за изведување на неколку прашалници беше потребно да се користи оваа функција, специфично користејќи го HiveQL јазикот во Hive алатката на Hadoop.

Во продолжение се дадени сите петнаесет прашалници и нивната имплементација во четирите бази на податоци, Amazon DynamoDB, Apache Cassandra, MongoDB, Neo4j.

| Прашалник | Amazon DynamoDB | Apache Cassandra | MongoDB | Neo4j |
|---|---|---|---|---|
| **Врати го вкупниот број на луѓе кои биле физички активни, но имале потешкотии во движењето и имаат срцево заболување.** | select count(*) from heartdata where PhysicalActivity = 'Yes' and DiffWalking = 'Yes' and HeartDisease = 'Yes' | select count(1) from data.hdata where heartdisease = 'Yes' and physicalactivity = 'Yes' and diffwalking = 'Yes' ALLOW FILTERING; | db.data.explain().find({ $and : [ { PhysicalActivity : 'Yes'}, { DiffWalking : 'Yes'}, { HeartDisease : 'Yes'} ]}).count() | MATCH(p:Person) WHERE p.PhysicalActivity = "Yes" AND p.DiffWalking = "Yes" AND p.HeartDisease ="Yes" RETURN COUNT(p) |

| | | | | |
|---|---|---|---|---|
| **Колкав е бројот на лица на возраст од 18 до 24 години кои имаат срцево заболување, а пушат цигари?** | select * from heartdata where HeartDisease = 'Yes' and AgeCategory = '18-24' and Smoking = 'Yes' | select count(1) from data.hdata where heartdisease = 'Yes' and agecategory = '18-24' and smoking = 'Yes' ALLOW FILTERING; | db.data.explain().find({ $and : [ { AgeCategory : '18-24'}, { Smoking : 'Yes'}, { HeartDisease : 'Yes'} ]}).count() | MATCH(p:Person) WHERE p.AgeCategory = "18-24" AND p.HeartDisease ="Yes" AND p.Smoking = "Yes" RETURN COUNT(p) |
| **Колкав е бројот на лица на возраст над 80 години кои имаат мозочен удар и срцево заболување?** | select * from heartdata where HeartDisease = 'Yes' and AgeCategory = '80 or older' and Stroke = 'Yes' | select count(*) from data.hdata where heartdisease = 'Yes' and agecategory = '80 or older' and stroke = 'Yes' ALLOW FILTERING; | db.data.explain().find({ $and : [ { AgeCategory : '80 or older'}, { Stroke : 'Yes'}, { HeartDisease : 'Yes'} ]}).count() | MATCH(p:Person) WHERE p.AgeCategory = "80 or older" AND p.HeartDisease ="Yes" AND p.Stroke = "Yes" RETURN COUNT(p) |
| **Прикажи ги сите информации за лицата кои немале мозочен удар и пиеле алкохол (над 14 пијалаци во текот на една недела за мажи и 7 за жени) и притоа немаат срцево заболување.** | select * from heartdata where HeartDisease = 'No' and AlcoholDrinking = 'Yes' and Stroke = 'No' | select * from data.hdata where heartdisease = 'No' and alcoholdrinking = 'Yes' and stroke = 'No' ALLOW FILTERING; | db.data.explain().find({ $and : [ { HeartDisease : 'No'}, { Stroke : 'No'}, { AlcoholDrinking : 'Yes'} ]}) | MATCH(p:Person) WHERE p.HeartDisease = "Yes" AND p.AlcoholDrinking ="Yes" AND p.Stroke = "No" RETURN p |
| **Врати го бројот на лица кои пијат алкохол, пушат цигари ( над 100 цигари во целиот живот), се на возраст од 65 години до 69** | select * from heartdata where Smoking = 'Yes' and AlcoholDrinking = 'Yes' and Stroke = 'Yes' | select count(*) from data.hdata where smoking = 'Yes' and alcoholdrinking = 'Yes' and stroke = 'Yes' ALLOW FILTERING; | db.data.explain().find({ $and : [ { AgeCategory : '65-69'}, { Smoking : 'Yes'}, { AlcoholDrinking : 'Yes'}, { Stroke : 'Yes'} ]}).count() | MATCH(p:Person) WHERE p.AlcoholDrinking ="Yes" AND p.Smoking = "Yes" AND p.Stroke = "Yes" AND p.AgeCategory = |

| | | | | |
|---|---|---|---|---|
| **години и притоа доживеале мозочен удар.** | | | | "65-69" RETURN COUNT(p) |
| **Најди го најстариот човек кој има рак на кожата, а притоа пуши цигари и има најголем индекс на телесна маса.** | select * from heartdata where BMI between 1 and 94.85 and Skincancer = 'Yes'and Smoking = 'Yes' and Heartdisease in ('Yes', 'No') order by BMI desc limit 1 | select * from data.hdata where skincancer = 'Yes'and smoking = 'Yes' and heartdisease in ('Yes', 'No') order by bmi desc limit 1 allow filtering; | db.data.aggregate([{ $match : { 'SkinCancer' : 'Yes', 'Smoking' : 'Yes'}},{ $sort : { AgeCategory : -1 , BMI : -1}},{ $limit : 1 }]).explain() | MATCH(p:Person) WHERE p.AgeCategory = "80 or older" AND p.SkinCancer = "Yes" AND p.Smoking = "Yes" RETURN p ORDER BY p.BMI DESC LIMIT 1 |
| **Прикажи го индексот на телесна маса на лицата кое имало проблеми со менталното здравје барем 20 дена во месецот, спиеле повеќе од 12 часови и пиеле алкохол.** | select BMI from heartdata where MentalHealth >= 20 and SleepTime > 12 and AlcoholDrinking = 'Yes' | select bmi from data.hdata where mentalhealth >= 20 and sleeptime > 12 and alcoholdrinking = 'Yes' allow filtering; | db.data.aggregate([{ $match : { 'MentalHealth' : {$gt : 20},'SleepTime' : {$gt : 12}, 'AlcoholDrinking' : 'Yes'}},{ $project :{ "_id" :0 , "BMI" :1 }}] | MATCH(p:Person) WHERE p.MentalHealth >=20 AND p.SleepTime >12 AND p.AlcoholDrinking = "Yes" RETURN p.BMI |
| **Спореди го просечниот индекс на телесна маса на мажите и жените кои имаат дијабетис, а притоа пијат алкохол и имаат потешкотии во одењето.** | Во HiveQL: select avg(BMI) from heartdata where Diabetic = 'Yes' and DiffWalking = 'Yes' and AlcoholDrinking = 'Yes' group by Sex | -За да се изврши овој прашалник потребно беше да создадам нова табела со примарен клуч атрибутот Sex и секундарни клучеви атрибутите: AlcoholDrinking, DiffWalking, Diabetic . | db.data.aggregate([{ $match : {'Diabetic':'Yes','AlcoholDrinking':'Yes','DiffWalking':'Yes'} },{ $group: { "_id" : '$Sex', "BMI" : { $avg : "$BMI"}}}]) | MATCH(p:Person) WHERE p.Diabetic = "Yes" AND p.AlcoholDrinking = "Yes" AND p.DiffWalking = "Yes" RETURN p.Sex, AVG(p.BMI) |

| | | | | |
|---|---|---|---|---|
| | | select bmi from data.sexgroupeddata where diabetic ='Yes' and alcoholdrinking = 'Yes' and diffwalking = 'Yes' group by sex allow filtering; | | |
| **Колку од луѓето во последните 30 дена имале проблеми со менталното здравје, кои имале физички потешкотии до 10 дена и доживеале срцев удар.** | select count(*) from heartdata where MentalHealth = 30 and PhysicalHealth < 10 and HeartDisease = 'Yes' | select count(*) from data.hdata where mentalhealth = 30 and physicalhealth < 10 and heartdisease = 'Yes' allow filtering; | db.data.aggregate([{ $match : { 'MentalHealth' : 30,'PhysicalHealth' : {$lt : 10}, 'HeartDisease' : 'Yes'}},{$count : "total"}]) | MATCH(p:Person) WHERE p.MentalHealth =30.0 AND p.PhysicalHealth <= 10 AND p.HeartDisease = "Yes" RETURN COUNT(p) |
| **Прикажи ја расата и индексот на телесна маса на најстарото лице кое е физички активно, генералното здравје му е одлично, нема физички проблеми ниту еден ден во месецот, а има индекс на телесна маса во граници на нормалата од 18.5 до 24.9.** | select Race, BMI from heartdata where AgeCategory = '80 or older' and PhysicalActivity = 'Yes' and GenHealth = 'Excellent' and PhysicalHealth = 0 and BMI between 18.5 and 24.9 limit 1 | select race, bmi from data.hdata where agecategory = '80 or older' and physicalactivity = 'Yes' and genhealth = 'Excellent' and physicalhealth = 0 and bmi >= 18.5 and bmi <= 24.9 limit 1 allow filtering; | db.data.aggregate([{ $match : { 'PhysicalHealth' : {$eq : 0},'GenHealth':'Excellent','AgeCategory' : '80 or older', 'BMI' : {$gt : 18.5, $lt: 24.9}}},{ $project :{ "_id" :0 , "BMI" :1, "Race": 1 }},{ $sort :{"BMI" : 1}},{ $limit : 1}]) | MATCH(p:Person) WHERE p.AgeCategory = "80 or older" AND p.PhysicalActivity = "Yes" AND p.GenHealth = "Excellent" AND p.PhysicalHealth = 0 and p.BMI >= 18.5 AND p.BMI <=24.9 RETURN p.Race, p.BMI LIMIT 1 |

| | | | | |
|---|---|---|---|---|
| **Прикажи ги деталните информации за испитаниците кои имале мозочен удар, срцево заболување, а притоа немаат проблеми со движењето и се физички активни иако имаат проблеми со менталното здравје барем 5 дена во месецот и го класифицираат своето генерално здравје како лошо или fair .** | select * from heartdata where Stroke = 'Yes' and HeartDisease = 'Yes' and DiffWalking = 'No' and PhysicalActivity = 'Yes' and MentalHealth >=5 and GenHealth = 'Poor' or GenHealth = 'Fair' | -нова табела со примарен клуч GenHealth<br>select * from data.genhealthdata where stroke = 'Yes' and heartdisease = 'Yes' and diffwalking = 'No' and physicalactivity = 'Yes' and mentalhealth >=5 and genhealth in ('Poor', 'Fair') allow filtering; | db.data.aggregate([{ $match : { 'MentalHealth' : {$gt : 5},'GenHealth' : {$in : ['Poor','Fair']},'Stroke' : 'Yes', 'HeartDisease':'Yes', 'DiffWalking' : 'No', 'PhysicalActivity':'Yes'}}]) | MATCH(p:Person) WHERE p.Stroke = "Yes" AND p.HeartDisease = "Yes" AND p.DiffWalking = "No" AND p.PhysicalActivity = "Yes" AND p.MentalHealth >= 5 AND p.GenHealth = "Poor" OR p.GenHealth = "Fair" RETURN p |
| **За секоја старосна група најди го минималниот, максималниот и просечниот број на денови во кои лицата имале ментално растројство, а притоа го оцениле своето генерално здравје специфично како добро, многу добро или одлично.** | select min(MentalHealth), max(MentalHealth), avg(MentalHealth) from heartdata where GenHealth in ('Good', 'Very good', 'Excellent') group by AgeCategory; | Во оваа база на податоци потребно е условот по кој пребарувате дали ја има посакуваната вредност да биде дел од групирањето ! | db.data.aggregate([{ $match: {$or: [{'GenHealth' : 'Good'}, {'GenHealth' : 'Very Good'},{'GenHealth':'Excellent'}]}},{ $group : {_id : '$AgeCategory', maxMentalHealth : { $max : "$MentalHealth"}, averageMentalHealth : {$avg : '$MentalHealth'}, minMentalHealth : {$min : '$MentalHealth'}}}]) | MATCH(p:Person) WHERE p.GenHealth = "Excellent" OR p.GenHealth = "Very Good" OR p.GenHealth = "Good" RETURN p.AgeCategory , MIN(p.MentalHealth), MAX(p.MentalHealth), AVG(p.MentalHealth)<br><br>MATCH(p:Person) WHERE |

| | | | | |
|---|---|---|---|---|
| **За секоја старосна група најди го минималниот, максималниот и просечниот број на денови во кои лицата имале ментално растројство, а притоа во групирањето додаде го и делот за оценката на своето генерално здравје специфично како добро, многу добро или одлично.** | select min(MentalHealth), max(MentalHealth), avg(MentalHealth) from heartdata where GenHealth in ('Good', 'Very good', 'Excellent') group by AgeCategory, GenHealth | select min(mentalhealth), max(mentalhealth), avg(mentalhealth) from data.genhealthdata where genhealth in ('Good', 'Very good', 'Excellent') group by genhealth, agecategory; | db.data.aggregate([{ $match: {$or: [{'GenHealth' : 'Good'}, {'GenHealth' : 'Very Good'},{'GenHealth':'Excellent'}]}},{ $group : {_id : {"age" : '$AgeCategory', "genhalth":'$GenHealth'}, maxMentalHealth : { $max : "$MentalHealth"}, averageMentalHealth : {$avg : '$MentalHealth'}, minMentalHealth : {$min : '$MentalHealth'}}}]); | p.GenHealth = "Excellent" OR p.GenHealth = "Very Good" OR p.GenHealth = "Good" RETURN p.AgeCategory , p.GenHealth, MIN(p.MentalHealth), MAX(p.MentalHealth), AVG(p.MentalHealth) |
| **Колку е вкупниот број на лица кои спијат во опсег од 7 до 9 часови дневно, имаат индекс на телесна маса во граници на нормалата од 18.5 до 24.9, се физички активни, немаат проблеми со физичкото здравје и менталното здравје, немаат проблеми со движењето, не пушат, не пијат** | Select count(*) from heartdata where SleepTime between 7 and 9 and BMI between 18.5 and 24.9 and PhysicalActivity = 'Yes' and PhysicalHealth = 0 and MentalHealth = 0 and DiffWalking = 'No' and Smoking = 'No' and AlcoholDrinkin | select count(*) from data.hdata where sleeptime >= 7 and sleeptime <=9 and bmi >= 18.5 and bmi <= 24.9 and physicalactivity = 'Yes' and physicalhealth = 0 and mentalhealth = 0 and diffwalking = 'No' and smoking = 'No' and alcoholdrinking = 'No' and heartdisease = 'No' and asthma = 'No' and skincancer = 'No' allow filtering; | db.data.aggregate([{ $match : { 'SleepTime' : {$gt : 7, $lt : 9},'BMI' : {$gt : 18.5, $lt: 24.9},'PhysicalHealth': { $eq : 0},'MentalHealth': {$eq : 0}, 'HeartDisease':'No','SkinCancer':'No','Asthma':'No','Smoking':'No','AlcoholDrinking':'No', 'DiffWalking' : 'No', 'PhysicalActivity':'Yes'}}, {$count : "total"}]) | MATCH(p:Person) WHERE p.SleepTime >= 7 AND p.SleepTime <= 9 AND p.BMI >= 18.5 AND p.BMI <= 24.9 AND p.PhysicalActivity = "Yes" AND p.PhysicalHealth = 0 AND p.MentalHealth = 0 AND p.DiffWalking = "No" AND p.Smoking = "No" AND p.AlcoholDrinking = "No" AND |

| | | | | |
|---|---|---|---|---|
| **алкохол, немаат срцево заболување, немаат астма и немаат рак на кожата.** | g = 'No' and HeartDisease = 'No' and Asthma = 'No' and SkinCancer = 'No' | | | p.HeartDisease = "No" AND p.Asthma = "No" AND p.SkinCancer = "No" RETURN COUNT(p) |
| **Направи споредба на испитаниците според расата наоѓајќи го просечниот број на индекс на телесна маса само на испитаниците кои имаат нормален индекс на телесна маса.** | -Во HiveQL select Race, avg(BMI) from heartdata where BMI between 18.5 and 24.9 group by Race | -Нова табела во која атрибутот Race е примарен клуч select race, avg(bmi) from data.racedata where bmi >=18.5 and bmi <= 24.9 group by race allow filtering; | db.data.aggregate([{ $match: {'BMI':{$gt:18.5, $lt:24.9}}},{ $group : {_id : '$Race', avgNormalBMI : { $avg : "$BMI"}}}]) | MATCH(p:Person) WHERE p.BMI >= 18.5 AND p.BMI <=24.9 RETURN p.Race, AVG(p.BMI) |
| **Колку е вкупниот број на лица и минимум, максимум, просечно и вкупниот број на часови кои ги поминуваат во спиење испитаниците кои имаат проблеми со менталното здравје 30 дена во месецот.** | select count(*), min(SleepTime), max(SleepTime), avg(SleepTime), sum(SleepTime) from heartdata where MentalHealth = 30 | select count(*), min(sleeptime), max(sleeptime), avg(sleeptime), sum(sleeptime) from data.hdata where mentalhealth = 30 allow filtering; | db.data.aggregate([{ $match :{ 'MentalHealth': { $eq : 30}}},{ $group : {_id : '$MentalHealth', totalNumber: { $count : {} }, avgSleepTime : { $avg : "$SleepTime"}, minSleepTime :{$min : "$SleepTime"}, maxSleepTime : {$max : "$SleepTime"}, totalSleepTime: {$sum: "$SleepTime"}}}]); | 15. MATCH( p:Person) WHERE p.MentalHealth = 30.0 RETURN p.MentalHealth AS MentalHealth, COUNT(p) AS Total, MIN(p.SleepTime) AS MinSleepHours, MAX(p.SleepTime) AS MaxSleepHours, AVG(p.SleepTime) AS AvgSleepHours, |

| | | | | |
|---|---|---|---|---|
| | | | | SUM(p.SleepTime) AS TotalSleepHours |

Овие петнаесет прашања имплементирани во сите четири бази на податоци беа потребни за да направи детална анализа на перформансите на неструктурираните бази на податоци во продолжение. Многу клучен фактор во анализата на перформансите на овие неструктурирани бази на податоци е и внесот на податоците, како и начинот на отстранување на истите.

## 5.2. Анализа на перформансите на претставниците на неструктурираните бази на податоци според извршените прашања и добиените резултати

Оваа анализа е направена според неколку клучни метрики и тоа :

- начинот на користење на базите на податоци т.е. достапноста на ресурси и нивната инсталација,
- синтаксата и интуитивноста при имплементирање на прашалниците во секоја од базите на податоци,
- можноста за имплементација на различните агрегатни функции,
- времето потрошено за внес на податочното множество,
- времето потрошено за бришење на податоците од базата на податоци,
- времето потребно за извршување на прашалниците.

### Анализа на начинот на користење на базите на податоци, синтаксата, интуитивноста и можноста за имплементирање на прашалниците во секоја од базите на податоци

Неструктурираните бази на податоци од фамилијата т.е. key value pair на прв поглед изгледаат доста лесни за користење, со достапни ресурси и едноставен јазик за пишување на прашалниците. Но, реалноста е малку поразлична. Прв предизвик со кој се соочив при проучување на претставниците од овие бази на податоци беше достапноста на истите само за ограничен број на оперативни системи т.е. поголемиот број од нив беа достапни само за Linux оперативниот систем. Поради горенаведената причина, а воедно и поради се поголемата популарност на онлајн сервисите на компанијата Амазон, се одлучив да почнам со работа со Amazon DynamoDB. Оваа база на податоци, има своја онлајн верзија, но имаме и можност за нејзина локална инсталација и имплементација. Воглавно, јас ја користев

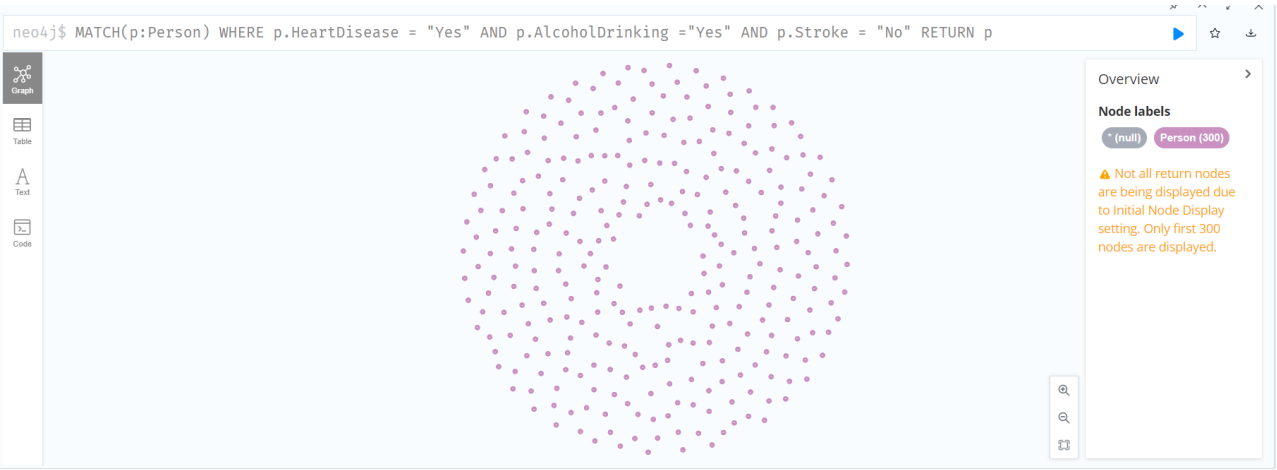
онлајн верзијата, но се соочив со неколку проблеми. Имено, онлајн верзијата на оваа база на податоци го нуди DynamoDB API за PartiQL, јазик компатибилен со SQL, но истата не ги подржува некои од агрегатите функции како: ограничување на бројот на вратени податоци, прикажување на минимална верност, на максимална вредност, сумирање, просечна вредност. Токму поради овие проблеми јас ја користев и локалната верзија на базата на податоци. За работа со оваа база на податоци користев и Hive алатка, која е дел од друг сервис т.е. од Hadoop и која во позадина работи на map-reduce концептот за да врати резултат. За работа со оваа алатка е потребен јазикот HiveQL, кој е многу сличен на PartiQL. HiveQL и Hive ги користев за имплементација на прашалниците кои во себе ја опфаќаа агрегатната функција за групирање. Групирањето во позадина, кај базите на податоци од типот на клуч вредност парови, е доста комплицирана операција поради потребата од пребарување низ примарните и секундарни клучеви за да го најдеме потребниот атрибут за групирање. Работата со оваа база на податоци беше доста покомплексна од другите бази од мој интерес. Воедно во оваа клуч вредност неструктурирана база на податоци можностите за скенирање на податоците се ограничени на примарните и секундарни клучеви, но истата нуди и неколку видови на филтрирање при пишување на обично прашање. Интерфејсот за работа со оваа база на податоци е доста едноставен и прегледен, а позитивна особина е дека за основите функции можете да ја користите само онлајн верзијата на PartiQL едиторот и не е потребно да трошите меморија на својот локален компјутер.

Сите други три бази на податоци, Apache Cassandra, MongoDB, Neo4j, не нудат онлајн верзии за извршување на прашалнци и истите потребно е да ги инсталирате и имплементиравте на својот локален компјутер. Но, начинот на имплементација е многу едноставен.

Apache Cassandra е навидум доста едноставна за користење поради нејзината голема сличност со релационите бази на податоци, поради нејзиниот т.н. column-oriented вид на неструктурирана база на податоци. Со релациони базни на податоци се среќававме многу често во последните четири години на факултетот, па работата и изучувањето на Apache Cassandra е доста едноставна. Но при имплементација на прашалниците во оваа база на податоци многу често се најдов во ситуација во која сфатив дека сепак работиме со неструктурирана база на податоци, која не го следи т.н. ACID модел на трансакции и која не е релациона база на податоци. Овој факт е особено воочлив при имплементацијата на прашалниците, од наш интерес, кои не го обработуваат главниот клуч и за кои потребно е да се направи нова табела. Базата само по себе не дозволува филтрирање, па ние тоа треба рачно да го овозможиме. Оваа база на податоци користи јазик за пребарување речиси идентичен како SQL.

MongoDB е т.н. document-oriented неструктурирана база на податоци со над 300 милиони преземања и истата е на врвот од листа за најкористени неструктурирани бази на податоци. Во мојот случај, оваа база на податоци беше наједноставна, најинтуитивна и со најлесна синтакса за имплементирање на прашалниците. Истата е многу едноставна за употреба и доколку немате познавања од бази на податоци, многу едноставно можете да се снајдете со оваа неструктурирана база на податоци поради тоа што нејзиниот јазик за пребарување и пишување на податоци го следи човековиот говор и размислување и на некој начин како зборуваме така и ги пишуваме работите во оваа база. Во MongoDB немате потреба на податоците да додавате никакви ознаки т.е. лабели, што е случај кај јазлите во граф базите, немате потреба од клучеви, што од друга страна е многу битно за клуч вредност базите на податоци, како и за базите на податоци базирани на колони. Оваа база на податоци нуди многу дополнителни алатки, како и своја онлајн верзија, MongoDB Atlas, со чија помош можете да ги чувате своите податоци во облак по ваш избор, но нема онлајн конзола. Локалната конзола потребно е да ја поврзете со MongoDB Atlas, доколку сакате таму да ви се зачувани податоците.

Како последна база на податоци која што ја обработив, а е претставник од видот на т.н. Graph-based неструктурираните бази на податоци е се популарната и револуционерна база Neo4j. Neo4j нуди едноставен интерфејс за работа со податоците и воедно нуди неколку начини на прикажување на резултатите при извршување на прашања врз податоците со помош на јазикот Cypher. Оваа база на податоци нуди многу корисна и богата документација, како и можност за едноставна интеграција со многу голем број на програмски јазици како .Net, Java, Node.js, Python. Neo4j ја нуди алатката Neo4j пребарувач која што се синхронизира со локалната верзија на оваа база на податоци и со Neo4j серверот и ни нуди можност да ги пристапиме нашите податоци од било каде. Јазикот за пишување на прашалниците, Cypher е многу интуитивен и лесен за пишување. Креаторите на оваа база на податоци се труделе јазикот за пребарување да биде логички и лесно разбирлив. Ова сепак е малку поразлична база на податоци која сама по себе нуди можност за пишување на покомплицирани прашалници, за испитување на врските помеѓу податоците на едно податочно множество, како и за додавање на параметри како на јазлите, така и на релациите меѓу нив т.е. на врските. На едноставен начин можете да означите јазли, да имплементирате некоја од агрегатните функции, како и да ги исфилтрирате резултатите. Neo4j нуди многу моќна визуелизација на резултатите, со чија помош може визуелно да ја увидиме формата на графот кој што го формираат нашите податоци, но доколку сакаме може да ги видиме резултатите и во JSON формат.




**Слика 43 „Изглед на визуелен приказ на резултат од извршен прашалник во Neo4j“**

## Анализа на времето потрошено за внес и отстранување на податоците од податочното множество

Сите четири бази на податоци од мој интерес се претставници од неструктурираните бази на податоци и согласно ова немаат однапред зададена шема на податоците. Но, овие претставници се од различни подвидови базирани на: клуч вредност парови (Amazon DynamoDB), колони (Apache Cassandra), документи (MongoDB), граф (Neo4j). Поради нивната заедничка главна категорија сите овие бази на податоци имаат свои сличности, како што се овозможувањето на евентуална или непосредна конзистентност, репликација на податоците за гарантирана достапност, методи за поделба т.е. партиционирање на податоците (Распарчување т.е. Sharding за Amazon DynamoDB, Apache Cassandra, MongoDB и Neo4j Fabric за Neo4j), конкурентност итн.

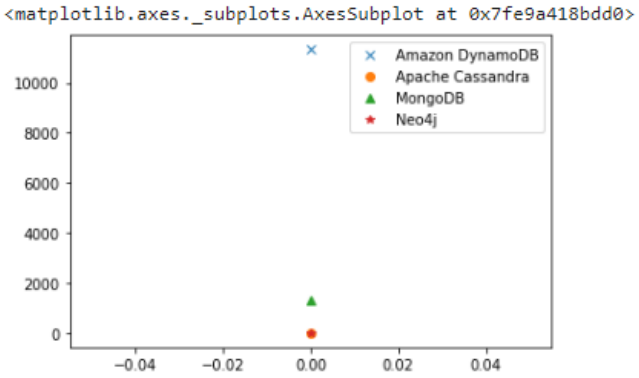
Но, од друга страна имаат и неколку поважни разлики кои ги класифицираат во различни подвидови како што се потребата за однапред дефиниран клуч кај Amazon DynamoDB и Apache Cassandra, како и дефинирањето на видот на јазлите т.е. означувањето на јазлите кај Neo4j и зачувувањето во т.н. JSON формат кај MongoDB. Поради овие неколку карактеристики имаме значителни разлики во времето потребно за внес на податоците во секоја од базите.

Според резултатите кои се добиени со мерење на времето потребно за внес на податочното множество, кое ги обработува клучните предизвикувачи на срцево заболување во секоја од базите, прикажани на сликите 44 и 45, можеме да увидиме дека Apache Cassandra и Neo4j имаат речиси идентично потрошено време за внес на податоците, а притоа истото е и значително помало наспроти Amazon DynamoDB и MongoDB. Но, потребно е да се нагласи дека податоците во Apache Cassandra и Neo4j беа внесени преку

конзола, т.е. директно преку графичките интерфејси на двете бази. А, за внес на податоците во MongoDB беше користена онлајн верзијата на базата на податоци MongoDB Atlas и алатката MongoDB Compass, што значи дека целиот процес беше извршен онлајн и многу зависи од брзината на интернетот со кој располагав во тој момент. Слична е ситуацијата и со Amazon DynamoDB. Со оглед на тоа дека оваа база на податоци е дел од веб сервисите на компанијата Amazon, за внес на поголем број на податоци напишав скрипта во програмскиот јазик Python, што претставуваше поврзување со клиент од трета страна со базата на податоци и беше потребно да има конекција, како за извршување на скриптата, така и за внес на податоците во базата.

| | Amazon DynamoDB | Apache Cassandra | MongoDB | Neo4j |
|---|---|---|---|---|
| 0 | 11337s | 16.329s | 1320s | 3.44s |

**Слика 44 „Табеларен приказ на времето потрошено за внес на податоците во секоја од базите соодветно, во секунди“**



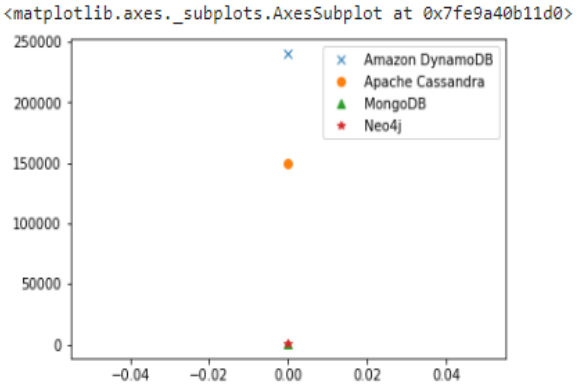

**Слика 45 „Графички приказ за споредба на вредностите за времето на внес на податочното множество на секоја од базите една наспроти друга“**

За да можеме да донесеме валиден заклучок за брзината на извршување на секој вид на прашалници врз овие четири претставници на неструктурираните бази на податоци потребно е да го увидиме и времето потребно за отстранување на податоците од базите. Податочното множество од наш интерес содржи податоци од 319 795 испитаници и согласно ова, очекуваме дека е потребно време за да се отстранат податоците. Според резултатите кои ги добив, а кои се визуелно прикажани на сликите 46 и 47, можеме да увидиме дека Neo4j и MongoDB се карактеризираат со значително помало време на извршување на оваа операција во споредба со Amazon DynamoDB и Apache Cassandra. Amazon DynamoDB одзема доста поголемо време за извршување на оваа операција, но

оваа база на податоци ни овозможува во рок од 48 часа да ги повратиме избришаните податоци доколку сме направиле некоја грешка.

| | Amazon DynamoDB | Apache Cassandra | MongoDB | Neo4j |
|---|---|---|---|---|
| 0 | 240000ms | 149724ms | 730ms | 778ms |

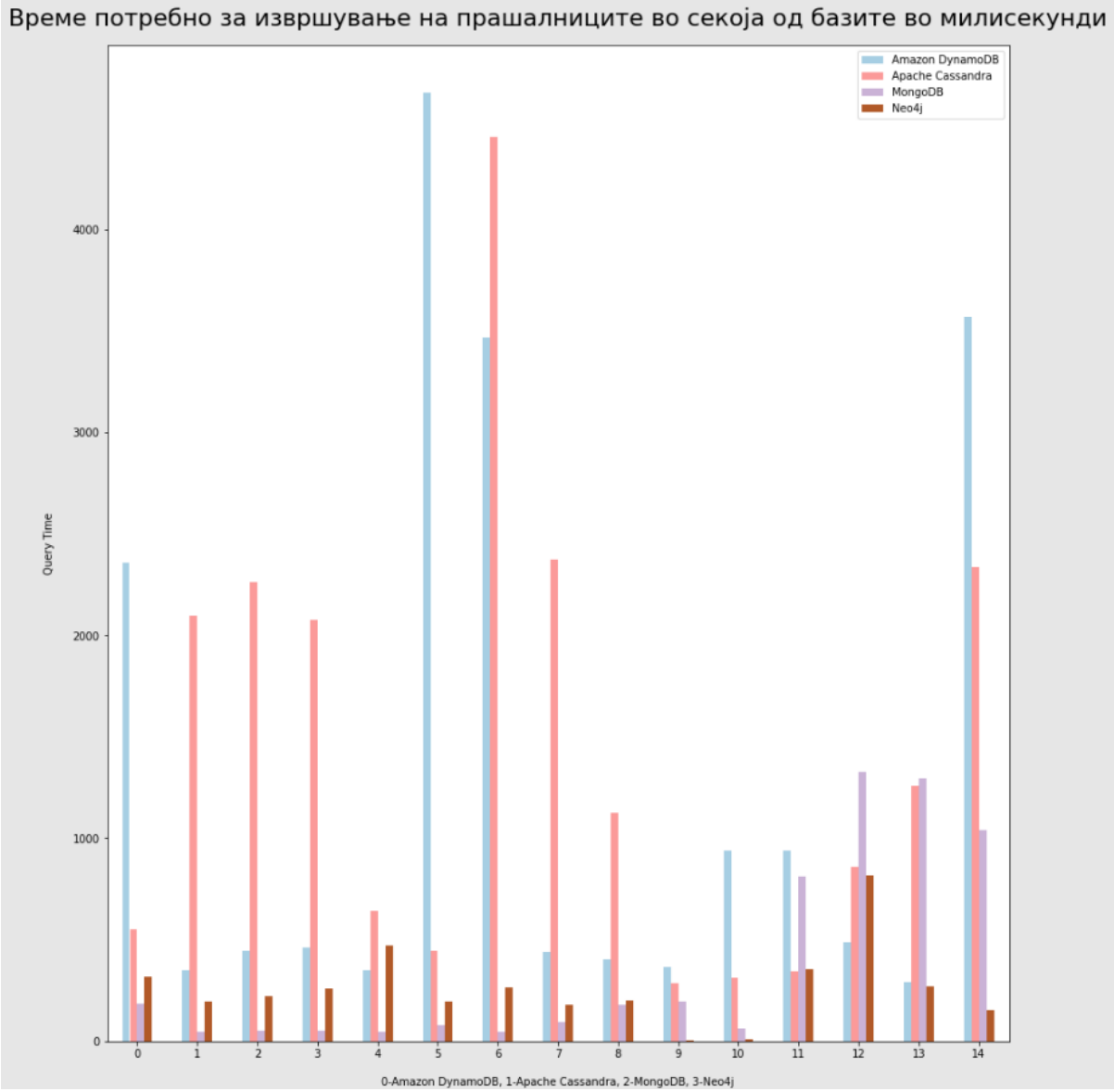
**Слика 46 „Табеларен приказ на времето потрошено за отстранување на податоците од секоја од базите соодветно, во милисекунди“**



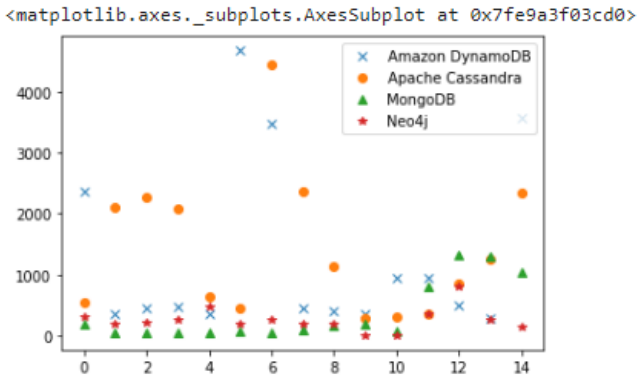

**Слика 47 „Графички приказ за споредба на вредностите за времето потребно за отстранување на податочното множество од секоја од базите на податоци, соодветно една наспроти друга“**

## Анализа на времето потребно за извршување на прашалниците

По направената анализа на резултатите кои ги обработуваа времињата на внесување и отстранување на податоците од четирите неструктурирани бази на податоци, време е да преминеме на следниот чекор т.е. да направиме комплетна анализа на времето потребно за извршување на прашалниците со различно ниво на тежина во секоја од базите претставници. За да добиеме чиста слика на времето потрошено во милисекунди за секој од прашалниците може да ги разгледаме и анализираме графиконите во продолжение.




**Слика 48 „Графички приказ на времето потрошено за извршување на секој прашалник во сите бази соодветно “**




**Слика 49 „Графички приказ на времето на извршување на секој прашалник во 4-те претставници на неструктурирани бази на податоци, означени со различна ознака дадена во легендата“**

Според добиените резултати, кои можеме да ги увидиме на хистограмите на сликите 48 и 49, можеме да заклучиме дека Neo4j има најмало време на извршување во покомплексните прашалници, кои во себе обработуваат неколку различни видови на агрегација како групирање, минимална вредност, максимална вредност, броење на единки, барање на сума. Од, друга страна MongoDB има помали вредности за поедноставните прашалници, како што се прашалниците од 0 до 6, кои во себе обработуваат неколку услови . Генерално, Amazon DynamoDB и Apache Cassandra одземаат значително повеќе време кога пребаруваат низ атрибути кои не се дел од примарните или секундарните клучеви. Но, за групирањето по примарен клуч , кое што е и единствен дозволен начин на групирање во Apache Cassandra, а за кој што е потребна Hive операција со map- reduce во Amazon DynamoDB, истите одземаат помалку време од MongoDB.

Според цело ова истражување, а и според многу истражувања објавени на интернет, за генерална употребна MongoDB е најбрзата база на податоци, меѓу овие четири претставници. Neo4j е најреволуционерната база на податоци која ни нуди доста широк спектар на операции и со која може да се зачувуваат и обработуваат и врските меѓу јазлите. Поради својата ориентираност и базираност на колони Apache Cassandra е моќна и воедно брза база на податоци за пребарување и групирање според колоната т.е. атрибутот кој е примарен клуч. И за крај, Amazon DynamoDB покажува најслаби перформанси во моето истражување и е најкомплицираната база на податоци за употреба меѓу овие четири претставници. Воедно за користење на оваа база на податоци, потребно е да се претплатите на веб страната на AWS т.е. Amazon Web Services.

По направената детална анализа, според мене, наједноставна, најинтуитивна и генерално најбрза неструктурирана база меѓу Amazon DynamoDB, Apache Cassandra, MongoDB и Neo4j е MongoDB, која што воено е и најкористена неструктурирана база на податоци.

# 6. Модел за предвидување на срцево заболување

XGBoost т.е. Extreme Gradient Boosting е алгоритам за машинско учење базиран на алгоритамот дрво на одлука и засилен со помош на т.н. gradient boosting рамка. Во проблемите кои на влез примаат податоци кои се целосно неструктурирани како слики и текст, невронските мрежи покажуваат супериорни резултати. Но, во мојот случај природата на податоците во множеството може да се класифицира како малку по структурирана, па алгоритмите базирани на дрво на одлука се подобар избор. Овој алгоритам е создаден во 2016 година, под закрила на универзитетот во Вашингтон и поради својот отворен код секојдневно научниците од областа на обработка на податоците, го надоградуваат и усовршуваат ден денес. Овој алгоритам е комбинација на хардверска и софтверска оптимизација за добивање на супериорни резултати за кратко време. Истиот имплементира паралелно извршување, кастрење на дрва, регулација, вкрстена валидација, пронаоѓање на оптимални точки т.е. Weighted Quantile Sketch.

Моделот за предвидување на срцево заболување е создаден во програмскиот јазик Python, со помош на библиотеки од машинско учење како sklearn, xgboost, scipy, pandas, matplotlib, инт.

Пред да го изградам моделот, атрибутите кои имаа само две уникатни текстуални вредности ги претворив во нумерички бинарни вредности т.е. во 1 и 0. А, атрибутите со поголем број на текстуални податоци, ги енкодирав со LabelEncoder функцијата која е дел од Python библиотека sklearn. За стандардизација на податоците кои ми беа потребни за предвидување на колоната срцево заболување, т.е. за да добијат истите стандардна нормална дистрибуција, ја користев функцијата StandardScaler. По овој чекор, податоците беа подготвени за поделба во две категории: тренинг и тест. За поделба на податочното множество во овие две категории ја користев функцијата train_test_split со дефинирани параметри за големина на тест множеството од 20% од сите податоци и можност за мешање на податоците. По оваа поделба ги добивме следните четири поделби на податоците:

```
X train : (255836, 17) & X test: (63959, 17)
y train: (255836,) & y test: (63959,)
```

**Слика 50 „Поделба на податочното множество за тренирање и тестирање на моделот за предвидување на срцево заболување“**

XGB моделот за предвидување на срцево заболување ги содржи следните важни параметри: логистичка регресија за бинарна класификација, рата на учење од 0.5 за да не

се направи т.н. overfitting на моделот, поттикнувачки алгоритам т.н. Dart и 600 дрва на одлуки. Другите параметри се стандарди за овој алгоритам.

```
model_o = xgb.XGBClassifier( objective = 'binary:logistic',
                             base_score=0.5,
                             booster='dart',
                             reg_alpha=0,
                             reg_lambda=1,
                             n_estimators=600,
                             learning_rate=0.5
                            )
model_o.fit(X_train_o, y_train_o)
```

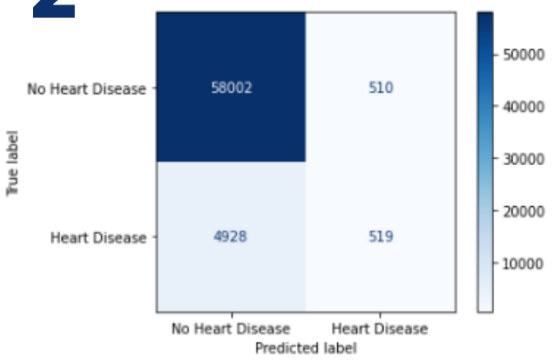
**Слика 51 „Моделот за преведување на срцево заболување“**

Моделот го тренирав на тренинг податоците, а го тестирав на тест податоците, при што добив точност од 91.497%. Деталните резултати се дадени на слика број 52 во продолжение.

1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.99 | 0.96 | 58512 |
| 1 | 0.50 | 0.10 | 0.16 | 5447 |
| accuracy | | | 0.91 | 63959 |
| macro avg | 0.71 | 0.54 | 0.56 | 63959 |
| weighted avg | 0.89 | 0.91 | 0.89 | 63959 |

2




3 Точност на предвидените тест податоци: 91.49767820009693

**Слика 52 „Резултати добиени по тестирањето на моделот за предвидување на срцево заболување“**

XGBoost е алгоритам кој што е дел од т.н. надгледувано учење. Овој алгоритам се обидува да ја предвиди точната вредност т.е. класа на примерокот со комбинирање на помали, послаби модели. Послабите модели кои ги користи алгоритамот во овој случај се дрва на регресија и секое дрво го мапира влезот во листови кои имаат континуирана вредност. XGBoost ја минимизира вредноста на функциите за регуларизација L1 и L2 кои користат т.н. конвексна функција на загуба т.е. базирани се на разликите помеѓу точната вредност за класата на примерокот и предвидената вредност на класата на примерокот. Тренинг процесот е итеративен т.е. со секоја итерација се додаваат нови дрва на одлука кои ја предвидуваат грешката на претходните дрва и потоа добиениот резултат од оваа итерација

се комбинира со резултатот од претходните итерации за да се добие конечен предвиден резултат. Воедно, наречен е и алгоритам кој користи постепено зголемување т.е. gradient boosting затоа што користи и алгоритам за постепено намалување на загубата при додавање на нови модели т.е. дрва на одлуки со цел да се минимизира загубата. Овој модел е составен од т.н. CART дрва на одлука т.е. класификациски и регресиски дрва на одлука кои се поразлични од обилните дрва на одлука. CART дрвата на одлука во своите листови содржат вистински вредности за припадноста на примерокот на некоја од групите, а не содржат само една одлука како обичните дрва на одлука. По достигањето на максималната длабочина, во овој случај 6, финална одлука се донесува со претворање на резултатите во категории користејќи праг на класификација со вредност 0.5. На слика 52, дадени си неколку видови на резултати добиени со овој модел. Во делот 1, од слика 52 прикажан е класификациски извештај кој се користи со цел да се измери квалитетот на предвидувањата од моделот за класификација. Сите метрики се добиени со користење на четири параметри и тоа:

- Број на вистински позитивни вредности кој што претставува број на случаи во кои класата била предвидена како позитивна и навистина била позитивна т.е. лицето имало навистина срцево заболување и било предвидено дека има срцево заболување,
- Број на лажно позитивни вредности кој што претставува број на случаи во кои класата била предвидена како позитивна, но всушност била негативна т.е. лицето немало срцево заболување но било предвидено дека има,
- Број на вистински негативни вредности кој што претставува број на случаи во кои класата била предвидена како негативна и навистина била негативна т.е. лицето од интерес немало срцево заболување и било предвидено дека нема срцево заболување,
- Број на лажно негативни вредности кој претставува број на случаи во кои класата била предвидена како негативна, а всушност била позитивна т.е. класификаторот предвидел дека лицето нема срцево заболување, а всушност имало .

Првата метрика е прецизноста која што ја означува способноста на класификаторот да ја предвиди точната класа на примерокот. Според дел 1, слика 52, класификаторот точно предвидел дека лицето нема срцево заболување дури 92% од случаите, а од друга страна класификаторот предвидел точно само за 50% од случаите за лицата кои имале срцево заболување. На прв поглед вредност на прецизноста за лицата кои навистина имале срцево заболување изгледа мала, но само 8.6% од испитаниците во податочното множество имале срцево заболување, па оваа вредност за прецизноста е сосема доволна. Втората метрика е т.н. recall што ја претставува способноста на класификаторот да ја ги најде сите точни примероци за одредена класа. Се пресметува како сооднос помеѓу бројот на точно предвидени вредности за дадена класа и збирот на бројот на точно предвидени

вредности за дадена класа и бројот на вредности кои биле од класата од интерес, но биле класифицирани како дел од другата класа. И за оваа метрика, вредноста за класата 0, т.е. за класата „Нема срцево заболување“, е многу поголема од истата причина како и за прецизноста. Ф1 резултатот т.е. f1 score се користи само за споредба на класификациските модели, не и за глобална точност поради тоа што резултат добиен со формулата: F1 Score = 2*(Recall * Precision) / (Recall + Precision) претставува урамнотежена средина на двете претходни метрики. Најдобриот резултат на Ф1 е 1, а најлош 0. Па според ова, резултатот за класата „Нема срцеви заболувања“, 0.96, е многу добар, а за класата „Има срцево заболување“ значително полош. Сепак најважна метрика е точноста која се пресметува како однос помеѓу бројот на точни предвидувања со вкупниот број на предвидувања за било која од двете класи и изнесува 91.497%. Графички приказ на соодносот на точно предвидени класи, наспроти неточно е даден во дел 2, на слика 52. Сепак можеме да увидиме дека многу поголемиот број на испитаници кои немаат срцево заболување, значително влијае на класификаторот и за да се подобрат добиените метрики за прецизност, recall, f1 score може да се отстранат дел од испитаниците или да се направи т.н. oversampling, но оваа постапка значително ќе ни ја намали точноста на моделот.

# 7. Заклучок

Секоја нова технологија, нов пронајдок, ново истражување ни го подобрува животот, ни го збогатува знаењето и не научува нешто ново. Живеењето во 21 век си носи свои предизвици и има свои добри и лоши страни. Една главна карактеристика на овој век е големиот напредок на технологијата во сферата на информациските технологии и медицината. Како во светот, така и кај нас во нашата земја, на пазарот и во ординациите се појавуваат многу паметни уредни кои што влеваат нова надеж, спасуваат живот. Секој нов ден, новите стресови, нездравиот и пребрз начин на живот доведува до влошување на нашето здравје. Со помош на новите технологии, медицинските лица на лесен и едноставен начин прибираат голем број на информации за штетните навики кои ни го загрозуваат нашиот живот. Овие податоци, ни овозможуваат да увидиме работи за кои потајно сме свесни дека ги правиме, а се штетни и животно загрозувачки. Ова истражување е направено со две главни цели и тоа: да го поттикне читателот да размисли за своите штетни навики и да ги анализира перформансите на неструктурираните бази на податоци кои имаат потенцијал да станат водечки во иднина.

Со помош на деталната анализа на податочното множество, кое ги обработува клучните предизвикатели на срцево заболување, можевме да увидиме дека прекумерната тежина, пушењето, стресот и нарушеното ментално здравје, како и физичката не активност и нездравиот сон се главни предизвикувачи на срцево заболување. Проблемите со срцето се најчестата болест во денешницата која е индиректен предизвикувач на мозочни удари. За жал, многу од предизвикувачите на срцево заболување, се предизвикувачи и на други болест, како на пример дијабетес, астма, рак на кожата. И сето ова не наведува да застанеме за момент и да размислиме што се погрешно правиме. Прекумерното уживање во нездравата храна и нездравите задоволства кога се чувствуваме лошо, во најтешките мигови или пак во миговите на среќа и радост можеби ќе ни донесат задоволство, но на долг рок се само идни предизвикатели на тешки болести. За среќа, постоењето на податочни множества кои ги обработуваат овие тематики, како податочното множество во фокусот на ова истражување и многу други множества собрани од медицинскиот персонал ни овозможуваат да направиме детална анализа и да се обидеме да смениме нешто во нашиот начин на живеење. Вакви податоци на светско ниво има многу, но многу е важно да се нагласи дека и во нашата земја се спроведуваат вакви истражување. Конкретно во Република Северна Македонија постои центар за статистичка обработка на здравствените податоци, публицистика и едукација при Институтот за јавно здравје на Република Северна Македонија кој што има бази на податоци за процентот на заболени од рак, од шеќерната болест итн. Но, овие податоци не се достапни за поширокатa јавност. Ова истражување го

подржува мислењето на Томас А. Едисон кој рекол дека „Лекарот на иднината нема да даде лек, туку ќе го подучи својот пациент да се грижи за себе, за својата исхрана и за причините и превенцијата на болестите. “

Од друга страна, оваа дипломска работа има за цел да не поттикне да сфатиме дека многу е важен и квалитетот, а не само квантитетот на податоците. Овој дел можевме да го увидиме од моделот на машинско учење кој што е направен за предвидување на срцево заболување. Форматот на податоците е доста предизвикувачки, земајќи го во предвид фактот дека немаме специфична шема кај неструктурираните податоци. Но, сите четири претставници на едноставен начин се справуваат со ваквата карактеристика, со помош на едноставната синтакса која што ја имаат и која за разлика од релационите бази на податоци е доста по интуитивна. Основната идеја која што ја имав пред почеток на ова истражување беше да ги анализирам можностите кои што ни ги нудат најпопуларните претставници на неструктурираните бази на податоци. По направената анализа, можам да кажам дека простор за најголемо експериментирање и најнесекојдневно искуство ни нуди Neo4j како претставник на граф неструктурираните бази на податоци. Доста блиски, со речиси идентична синтакса како и во релационите бази на податоци се Apache Cassandra, претставникот на неструктурираните бази на податоци базирани на колони и Amazon DynamoDB, претставникот на неструктурираните бази на податоци со клуч вредност парови. Додека, MongoDB со право е најкористена неструктурирана база на податоци, со голема брзина на извршување на прашалници, едноставна синтакса, репликација на податоците, најголем број на клиенти за различни програмски јазици итн.

Овие четири претставници на неструктурираните бази на податоци се доста популарни меѓу многу гиганти во голем број на индустрии. Конкретно Amazon DynamoDB е база на податоци користена од 15 352 компании, Apache Cassandra од 8 366 компании, MongoDB од 53 983 компании и Neo4j од 950 компании. Иако релационите бази на податоци сеуште се најкористените бази на податоци на светско ниво, генерално поради својот ACID модел на трансакции и потребата од конзистентност и прецизност, сепак можеме да кажеме дека неструктурираните бази на податоци имаат светла иднина. Неструктурираните бази на податоци се користени од мега популарните компании и организации како NASA, EBay, Zurich, Adobe, German Centre for Diabetes Research, Volvo Cars, Microsoft, AstraZeneca, Cisco, NBC News, HP, Forbes , Toyota, Bosch, Google, Twitter, Netflix, Apple, Disney, Dropbox, Samsung, PayPal, Amazon, Airbnb, Zoom, Nike ,инт.

Овие компании ги употребуваа неструктурираните бази на податоци поради нивната лесна употреба за неколку клучни работи како: едноставно складирање информации за сесиите на веб-апликациите, можноста за зачувување на податоци без специфична шема, можноста за создавање на глобална продавница на кориснички профили т.е. можноста за создавање на една целосна слика за корисникот со помош на информациите кои ги има поставено на

различни сервиси, супериорните перформанси и брзина за манипулации со податоци со голем обем како внес, ажурирање, испорачување на податоците, искористувањето на технологиите во облак за да нема прекини за доставата на податоци, брзата промена во мобилните апликациите без големи промени во инфраструктурата на базата на податоци, лесното кеширање на податоците и интеграцијата со апликации од трета страна, за глобално дистрибуирани складишта на податоци, приспособливоста на наглото зголемување на корисници, итн. Исто така, неструктурираните бази на податоци се почесто се користат во светот од компании кои се занимаваат со обработка на големите податоци т.е. big data во реално време, како што е PayPal компанијата, која со обработка на информациите од своите корисници на дневна база открива измами, ги класифицира своите корисници се со цел да им овозможи подобро искуство прикажувајќи им персонализирани реклами, пронаоѓа повторливи шеми т.е. патерни и врски меѓу податоците за предвидување на идните трендови т.н. data mining. Неструктурираните бази на податоци се користат и од страна на компаниите кои нудат IoT т.е. Internet of Things уреди и сервиси како што се паметните телефони, часовници, сензори, асистенти, се со цел да можат на лесен начин да се синхронизираат информациите помеѓу истите и да се понуди брз одговор на корисникот. Други употреби на овие бази имаме од страна на компаниите на кои: им е потребно управување со содржини, развиваат мобилни апликации за голем број на корисници, кои целат кон збогатување на дигиталното искуство на своите корисници, итн.

Неструктурираните бази на податоци земаат свој замав и во нашата држава. Голем број на програмерски компании во последно време се насочуваат кон користење на овие бази за своите проекти. Но, генералната слика за нашата земја е дека овие бази не се многу застапени, податочните множества кои што ги поседува државата се зачувани во релациони бази на податоци.

Од мојата дипломска работа можеме да заклучиме дека постојат многу нови технологии, кои со секој нов ден се надградуваат и доусовршуваат и кои можат навистина да ни го олеснат и подобрат начинот на живеење. Оваа дипломска работа има за цел да го мотивира читателот да застане за момент и да сфати дека штетните навики може да го загрозат неговиот живот. Сепак како што рекол Стиф Џобс, ко-основачот на компанијата Apple,: „Тоа не е верба во технологијата. Тоа е верба во луѓето. Мислам дека најголемите иновации на 21 век ќе бидат кога ќе создадеме пресек помеѓу биологијата и технологијата. Ќе започнеме нова ера“.

# 8. Користена литература


[1] Cardiovascular Risk In Men - Why Is Heart Disease A Male Problem: https://www.newvictoria.co.uk/about-us/news-and-articles/cardiovascular-risk-in-men-why-is-heart-disease-a-male-problem

[2] Early Signs of a Heart Attack: https://health.clevelandclinic.org/women-men-higher-risk-heart-attack/

[3] Heart Health Benefits of Physical Activity: https://www.ucsfhealth.org/education/heart-health-benefits-of-physical-activity

[4] Obesity and overweight: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

[5] How Excess Weight Impacts Our Mental and Emotional Health: https://www.ncoa.org/article/how-excess-weight-impacts-our-mental-and-emotional-health

[6] List of Best and Most Popular NoSQL Database 2022: https://www.innuy.com/blog/list-of-best-and-most-popular-nosql-database-2022/#:~:text=MongoDB%2C%20considered%20the%20most%20popular,%2Doriented%20open%2Dsource%20database

[7] DynamoDB: https://aws.amazon.com/dynamodb/

[8] Working with Amazon DynamoDB, Developer guide: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/

[9] Working with Amazon DynamoDB and Apache Hive in-depth explanation: https://www.youtube.com/watch?v=Z5bLj-w4VDs

[10] Connecting to AWS DynamoDB using Boto3 and Python: https://towardsaws.com/connecting-to-aws-dynamodb-using-boto3-and-python-6e4774588d24

[11] Python – AWS DynamoDB – Load data with Boto3 using JSON file as input: https://itecnote.com/tecnote/python-aws-dynamodb-load-data-with-boto3-using-json-file-as-input/

[12] Create Your First Table by Using the DynamoDB Console - AWS Virtual Workshop: https://www.youtube.com/watch?v=soNG0n68spw

[13] Query and Manage DynamoDB Tables by Using Python - AWS Virtual Workshop: https://www.youtube.com/watch?v=Lx5jfh5OYmg

[14] AWS DynamoDB PartiQL | AWS hands on tutorial for PartiQL to manipuplate DynamoDB tables: https://www.youtube.com/watch?v=Y0uniyl5EoM

[15] How to use PartiQL with DynamoDB: https://www.youtube.com/watch?v=ZNadZuiARVc

[16] Introduction to Apache Cassandra - the “Lamborghini” of the NoSQL World: https://www.datastax.com/blog/introduction-to-apache-cassandra-the-lamborghini-of-the-nosql-world

[17] Installing Apache Cassandra on Windows in 2021: https://www.youtube.com/watch?v=hJxlkHafYsQ

[18] Cassandra Documentation: https://cassandra.apache.org/doc/latest/

[19] How to Install Cassandra on Windows 10: https://phoenixnap.com/kb/install-cassandra-on-windows

[20] Apache Cassandra - Tutorial: https://www.youtube.com/watch?v=s1xc1HVsRk0&list=PLalrWAGybpB-L1PGA-NfFu2uiWHEsdscD

[21] MongoDB Tutorial: https://www.youtube.com/watch?v=-0X8mr6Q8Ew

[22] MongoDB Tutorial: https://www.javatpoint.com/mongodb-tutorial
[23] MongoDB Documentation: https://www.mongodb.com/docs/
[24] How to connect MongoDB Compass to MongoDB Atlas: https://www.youtube.com/watch?v=vAHd7oV1uE0
[25] MongoDB Compass: Import JSON and CSV files: https://www.youtube.com/watch?v=IjAflHMkuzk
[26] Neo4j (Graph Database) Crash Course: https://www.youtube.com/watch?v=8jNPelugC2s
[27] How-To: Import CSV Data with Neo4j Desktop: https://neo4j.com/developer/desktop-csv-import/
[28] Neo4j official website: https://neo4j.com/
[29] Neo4j official documentation: https://neo4j.com/docs/
[30] The 5 Fastest NoSQL Databases Every Data Science Professional Should Know About: https://bangdb.com/blog/fastest-databases/
[31] Cassandra vs. MongoDB vs. Neo4j: https://db-engines.com/en/system/Cassandra%3BMongoDB%3BNeo4j
[32] Amazon DynamoDB vs. Cassandra vs. MongoDB: https://db-engines.com/en/system/Amazon+DynamoDB%3BCassandra%3BMongoDB
[33] XGBoost Algorithm: Long May She Reign!: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
[34] NoSQL Database Deployments: 10 Real-World Examples:https://www.eweek.com/database/nosql-database-deployments-10-real-world-examples/
[35] Beyond “Fast and Simple”: Top 5 Use Cases for NoSQL Database Technology: https://www.budaconsulting.com/fast-and-simple-use-cases-for-nosql-database-technology/
[36] Центар за статистичка обработка на здравствените податоци, публицистика и едукација: https://www.iph.mk/centar-za-statisticka-obrabotka-na-zdravstveni-podatoci-publicistika-i-edukacija/
[37] Претпроцесирање и детална анализа на податочното множество, Дипломска работа 183045: https://colab.research.google.com/drive/1-rDe48Kt6-8zQUdyFrcVeJSWErb-1uu-?usp=sharing
[38] Визуелизација на резултатите од мерење на перформансите на претставниците од неструктурираните бази на податоци, Дипломска работа, 183045: https://colab.research.google.com/drive/14_neffv5CVHF6DBmUYN7pjsXYeg6-ZPU?usp=sharing
[39] Модел за предвидување на срцево заболување, Дипломска работа 183045: https://colab.research.google.com/drive/1sjtx9E-DFcs7sDIqR6iEc1_ExP7ur1rx?usp=sharing