

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6488

**KLASIFIKACIJA SRČANIH RITMOVA IZ SNIMAKA
12-KANALNOG ELEKTROKARDIOGRAMA**

Marko Barišić

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6488

**KLASIFIKACIJA SRČANIH RITMOVA IZ SNIMAKA
12-KANALNOG ELEKTROKARDIOGRAMA**

Marko Barišić

Zagreb, lipanj 2020.

ZAVRŠNI ZADATAK br. 6488

Pristupnik: **Marko Barišić (0036505569)**

Studij: Računarstvo

Modul: Računarska znanost

Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Klasifikacija srčanih ritmova iz snimaka 12-kanalnog elektrokardiograma**

Opis zadatka:

U ovom završnom radu potrebno je proučiti i opisati značajke srčanih ritmova snimljenih pomoću 12-kanalnog elektrokardiograma. Završni rad treba koristiti javno dostupni skup podataka s međunarodnog izazova za klasifikaciju nekoliko srčanih ritmova CinC 2020. Na temelju dostupne literature u ovom području, cilj rada je razviti vlastiti algoritam koji će na temelju snimke 12-kanalnog elektrokardiograma klasificirati o kojem srčanom ritmu se radi. Pritom algoritam može koristiti postojeće metode strojnog učenja za klasifikaciju. Algoritam treba naučiti raspoznavati ritmove na dostupnom skupu podataka za učenje, a vrednovanje treba provesti na odvojenom, također dostupnom skupu podataka za testiranje. S obzirom da međunarodni izazov nema dostupne stvarne podatke o ritmovima za testni skup, u radu treba navesti procjenu točnosti algoritma koristeći više mjera za vrednovanje klasifikatora koristeći unakrsnu validaciju na skupu za učenje.

Rok za predaju rada: 12. lipnja 2020.

Zahvala

Zahvaljujem se svom mentoru izv. prof. dr. sc. Alanu Joviću na izvrsnom prijedlogu teme za ovaj završni rad. Kao i na savjetima i materijalima koje je podijelio samnom.

Također, zahvaljujem se svojoj obitelji na potpori i povjerenju koje mi oduvijek ukazuju.

Hvala vam!

Sadržaj

1. UVOD	1
2. SRCE I ELEKTROKARDIOGRAM	2
2.1. Srce	2
2.1.1. Anatomija srca	2
2.1.2. Rad srca.....	3
2.1.3. Sustav električne provodljivosti srca.....	4
2.2. 12-kanalni elektrokardiogram	5
2.2.1. Princip rada	5
2.2.2. Analiza snimke.....	8
2.3. Ritmovi i aritmije.....	9
3. ALGORITMI.....	14
3.1. Analiza signala snimke EKG-a.....	14
3.1.1. Pan Tompkinsov algoritam	14
3.1.2. Elgendijev algoritam	16
3.2. Strojno učenje	18
3.2.1. Stroj s potpornim vektorima.....	18
3.2.2. Logistička regresija	23
4. IMPLEMENTACIJA.....	25
4.1. Podaci	25
4.2. Programska izvedba	27
4.2.1. Tok programa.....	28
4.2.2. Izvedba Pan-Tompkinsovog algoritma	29
4.2.3. Izvedba Elgendijevog algoritma.....	32
4.2.4. Morfološka analiza.....	35
4.3. Vrednovanje i rezultati.....	36
5. ZAKLJUČAK.....	42
6. LITERATURA	43
SAŽETAK	45
ABSTRACT.....	45

1. Uvod

Prema podacima Svjetske zdravstvene organizacije kardiovaskularne bolesti najčešći su i najrašireniji su uzrok smrti na svijetu [1]. Kardiovaskularne bolesti uzrok su procijenjenih 17.9 milijuna smrti godišnje što predstavlja 31% svih smrti godišnje [1]. Elektrokardiogram je uređaj za analizu rada srca, jednostavan je za korištenje, opće dostupan i učinkovit. Pružanjem javnog pristupa bazama podataka snimaka elektrokardiograma došlo je do tehnološkog razvoja u analizi i detekciji značajki na snimkama elektrokardiograma.

U ovom završnom radu rješava se problem klasifikacije srčanih ritmova iz snimaka 12-kanalnog elektrokardiograma korištenjem algoritama za analizu signala i algoritama strojnog učenja za klasifikaciju. Potreba za razvijanjem takvog sustava dolazi iz velikog potencijala njegove primjene. Neke od mogućih primjena su klasifikacija srčanih ritmova u stvarnom vremenu korištenjem nosive tehnologije, klasifikacija srčanih ritmova u velikom broju snimaka elektrokardiograma, npr. u bolnicama i klasifikacija u vremenski dugim snimkama, npr. srčani holteri.

Rad je strukturiran na sljedeći način. U poglavlju *Srce i elektrokardiogram* proučava se područje kardiologije, rad srca, princip rada elektrokardiograma, značajke ispisa elektrokardiograma i objašnjenje ritmova i aritmija promatranih u ovom radu. U poglavlju *Algoritmi* proučava se područje algoritama za analizu signala te ekstrakciju značajki iz snimaka 12-kanalnog elektrokardiograma i algoritama strojnog učenja. U poglavlju *Implementacija* pregledava se implementacija sustava za klasifikaciju srčanih ritmova iz snimaka 12-kanalnog elektorkardiograma, korišteni algoritmi za ekstrakciju ključnih značajki signala, stvaranje podataka za učenje sustava te algoritmi strojnog učenja korišteni u implementaciji.

2. Srce i elektrokardiogram

Znanje iz područja kardiologije ključno je za ovaj završni rad, prvenstveno za razumijevanje zadatka i rezultata. Ono je također važno za razumijevanje i primjenu algoritama objašnjenih u poglavlju 3. Poznavanje anatomije i rada srca vrlo je bitno za biranje značajki za ekstrakciju. Razumijevanje elektrokardiograma vrlo je bitno za biranje značajki za ekstrakciju. U nastavku ovog poglavlja proučava se srce, anatomija i rad srca, te zapis EKG-a, princip njegovog snimanja te analize snimke. Većina navedenoga preuzeta je iz [2].

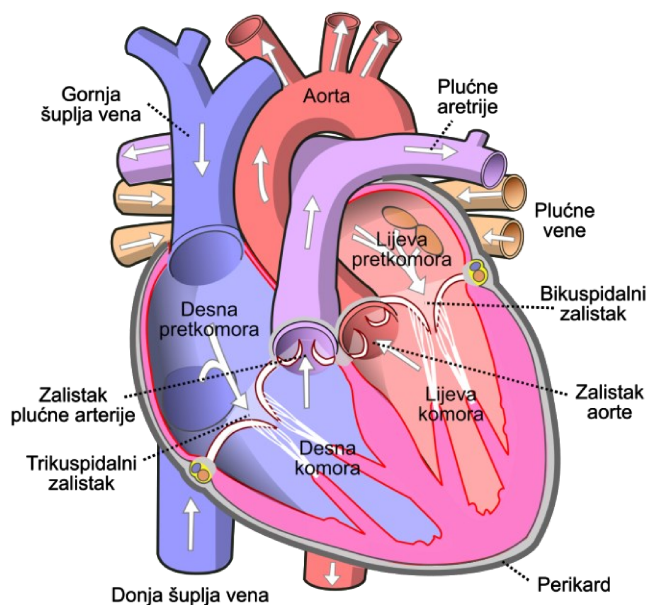
2.1. Srce

Srce je mišić veličine šake smješten u prsnoj šupljini, najčešće na lijevoj strani, između dva plućna krila. Srce je ključni dio krvožilnog sustava čovjeka u kojem se ponaša kao pumpa.

2.1.1. Anatomija srca

Srce je uzdužno podijeljeno na arterijski i venski dio. Arterijski dio obuhvaća lijevi dio srca, dok venski dio obuhvaća desni dio. Oba dijela su dalje podijeljena na još dva dijela, desnu i lijevu pretkljetku te desnu i lijevu kljetku. Drugi naziv za pretkljetku je atrij, a za kljetku ventrikul. Na slici 2.1 atriji su označeni kao pretkomore, a ventrikuli kao komore. Na prijelazu iz atrija u ventrikul nalaze se srčani zalisci. Bikuspidalni zalisci nalaze se na lijevoj, a trikuspidalni zalisci na desnoj strani. Srce je s ostatkom krvožilnog sustava povezano krvnim žilama. Vene su krvne žile koje dovode krv u srce, a arterije su krvne žile koje odvođe krv iz srca. Gornja i donja šuplja vena dovode deoksigeniranu krv iz cijelog tijela u desni atrij. Plućnim venama oksigenirana krv dolazi u lijevi atrij. Plućna arterija iz desnog

ventrikula odvodi krv u pluća. Iz lijevog ventrikula izlazi najšira krvna žila, aorta. Plućna arterija i aorta također imaju zaliske, oni su polumjesečastog oblika. Zalisci imaju važnu ulogu u radu srca, oni sprečavaju vraćanje krvi iz žila u srce.



Slika 2.1 – anatomija ljudskog srca [3]

2.1.2. Rad srca

Srce se stalno steže i opušta. Stezanje srca se naziva sistola, a opuštanje dijastola. Kao što je navedeno u prethodnom potpoglavlju, vene dovode, a arterije odvede krv iz srca. Postoje veliki i mali optok krvi, tj. plućni i sistemski optok krvi.

Mali optok krvi započinje sistolom, deoksigenirana krv plućnom arterijom odlazi u pluća. U plućima se ta krv oksigenira, oslobađa se ugljikov dioksid iz krvi te se krv obogaćuje kisikom. Iz pluća krv se vraća u lijevi atrij plućnim venama. Tada krv iz lijevog atrija odlazi u lijevi ventrikul od kuda se potiskuje u aortu.

Veliki optok krvi započinjem sistolom, oksigenirana krv aortom odlazi u cijelo tijelo. Iz aorte krv teče kroz sve manje i manje arterije. Konačno dolazi u kapilare kojima

teče i izmjenjuje plinove sa stanicama. Krv predaje stanicama kisik, a od njih preuzima ugljikov dioksid. Time oksigenirana krv postaje deoksigenirana krv te putuje venulama, malim i velikim venama. Konačno preko gornje i donje šuplje vene dolazi do desnog atrija. Tu se sistolom potiskuje u desni ventrikul. To je kraj velikog optoka krvi i ponovno počinje mali optok krvi.

Količina krvi, koja se u periodu trajanja jedne sistole potisne u aortu, naziva se udarni volumen srca. Udarni volumen srca u prosjeku iznosi 70 ml.

2.1.3. Sustav električne provodljivosti srca

Srce radi samostalno. Središte rada srca je sustav električne provodljivosti srca. Glavna uloga tog sustava je stvaranje i prijenos električnih impulsa ostatku stijenke srčanog mišića (miokarda). Ovaj proces stvara električnu energiju koju očitavaju elektrode EKG-a, više o tome u poglavlju 2.2. Dijelovi tog sustava su sinus-atrijski čvor (S-A čvor), atrio-ventrikularni čvor (A-V čvor), Hissov snop te Purkinjeova vlakna. Svi dijelovi protežu se miokardom.

S-A čvor nalazi se u desnom atriju. Ovaj dio sustava zadužen je za određivanje frekvencije otkucaja srca, ima ulogu glavnog održavatelja ritma rada (*pejsmejker*) srca. S-A čvor održava frekvenciju u rasponu od 60 do 100 otkucaja u minuti, u prosjeku frekvencija iznosi 70 otkucaja/min. Usporeni rad srca, manje od 60 otkucaja/min, naziva se sinusna bradikardija, dok se ubrzani rad srca, više od 100 otkucaja/min, naziva sinusna tahikardija.

A-V čvor slijedno je sljedeći u sustavu električne provodljivosti srca. Nalazi se na prijelazu iz desnog atrija u desni ventrikul. A-V čvor također služi kao sporedni pejsmejker srca, održava rad srca u rasponu od 40 do 60 otkucaja u minuti. Ukoliko S-A čvor ne pokrene električni impuls, A-V čvor preuzima ulogu glavnog pejsmejker srca.

Hissov snop služi za prijenos impulsa od A-V čvora nadalje. Nalazi se dijelom u zidu desnog atrija i u pregradi između ventrikula. Ima desnu i lijevu granu koje se granaju prema odgovarajućim ventrikulima. Hissov snop jedini je komunikacijski kanal između atrija i ventrikula. Purkinjeova vlakna granaju se iz Hissovog snopa. Ona su skup stanica koje izravno podražuju desni i lijevi ventrikul.

Stanice miokarda u mirovanju imaju negativni električni potencijal na membrani, otprilike od -70 do -90 mV. Natrij-kalijeva pumpa polako smanjuje negativni naboj unutar stanice te povećava električni potencijal. U jednom se trenutku naboj unutar i izvan stanice ujednače. To se naziva depolarizacijom. Tada ioni kalcija ulaze u stanice i događa se stezanje miokarda. Zatim se stanica nastoji vratiti u stanje mirovanja i to čini smanjenjem pozitivnog naboja unutar sebe. Konačno dolazi do repolarizacije stanica i događa se opuštanje miokarda. Dakle, depolarizacija se očituje stezanjem miokarda, dok se repolarizacija očituje opuštanjem miokarda. Depolarizacija i repolarizacija stanica miokarda prenose se sustavom električne provodljivosti srca.

2.2. 12-kanalni elektrokardiogram

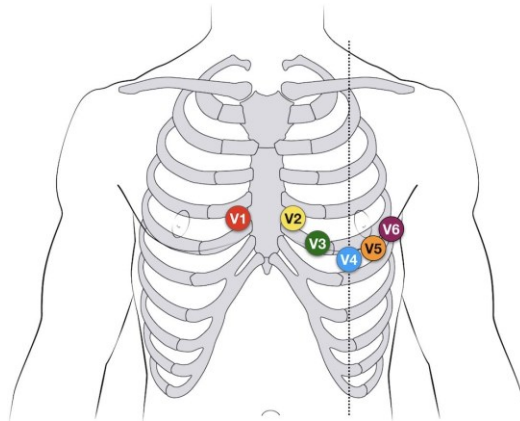
Elektrokardiograf je uređaj koji zapisuje električnu aktivnost srca u obliku valova elektrokardiograma. Elektrodama, kojima se spaja na pacijenta, detektira električnu aktivnost. U ovom poglavlju objašnjen je 12-kanalni EKG.

2.2.1. Princip rada

Elektrode EKG-a detektiraju električnu aktivnost srca koja se odvija ispod njih. Kada se pozitivno nabijeni električni impuls udaljava od elektrode, EKG ga zapisuje kao negativni val. Kada se takav električni impuls giba prema elektrodi, EKG ga zapisuje kao pozitivni val. Dakle, EKG pretvara električnu aktivnost detektiranu elektrodama

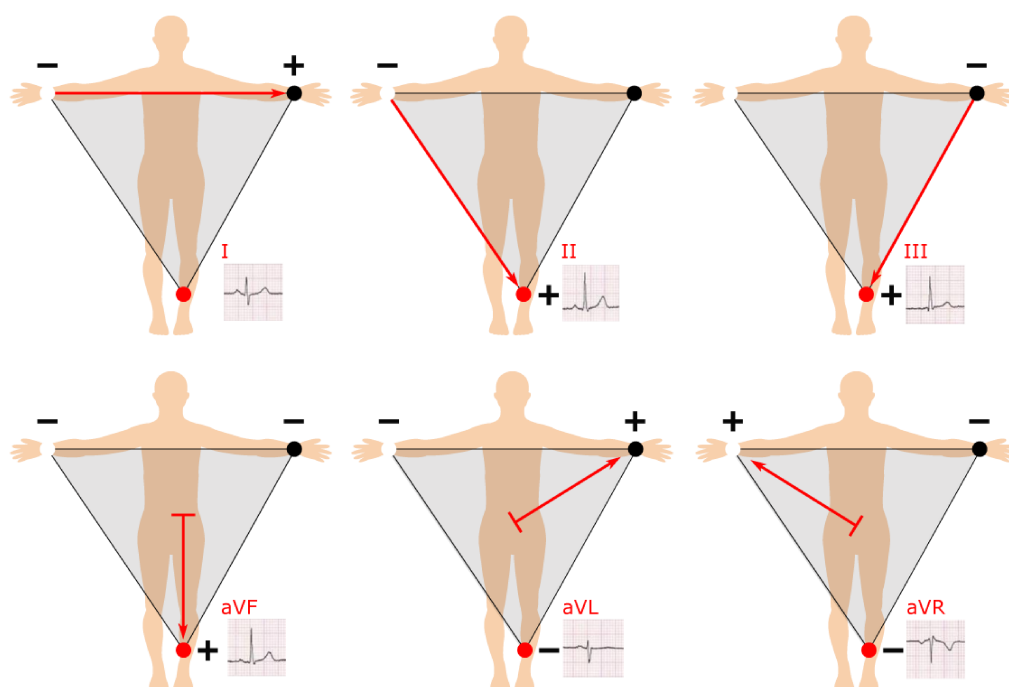
u valove. Kanale se može shvatiti kao perspektivu iz koje se promatra električna aktivnost srca.

12-kanalni EKG ima 10 elektroda. Četiri elektorde postavljaju se na udove, obje ruke i noge. Za ove elektorde nije bitan točan položaj na udu, već je ključno da su udaljene barem 10 cm od srca. Ostalih šest elektroda postavlja se kako je prikazano na slici 2.2.



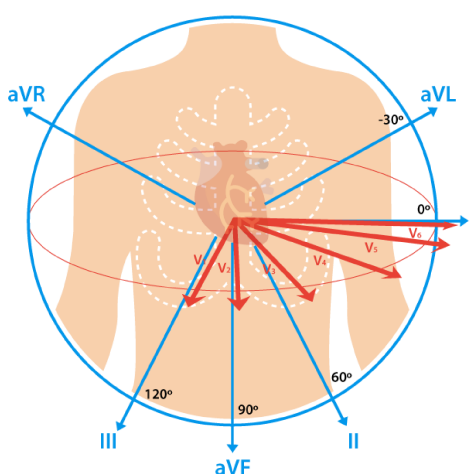
Slika 2.2 – položaj prekordijalnih elektroda [4]

Elektrode na rukama imenovane su RA, LA, RL, LL gdje R i L označavaju desno (R) i lijevo (L), a A i L označavaju ruku (A) i nogu (L). Elektroda RL se ne prikazuje na očitavanju EKG-a, već ona služi kao svojevrsni stabilizator za smanjenje smetnji pri mjerenju EKG-om. Ostale tri elektode čine točke Einthovenovog trokuta pomoću kojeg se tvori 6 kanala od njih 12. Ti kanali su I, II, III, aVF, aVL i aVR. Princip Einthovenovog trokuta prikazan je na slici 2.3. Stvoreni kanali promatraju električnu aktivnost srca na poprečnom presjeku srca.



Slika 2.3 – Einthovenov trokut [5]

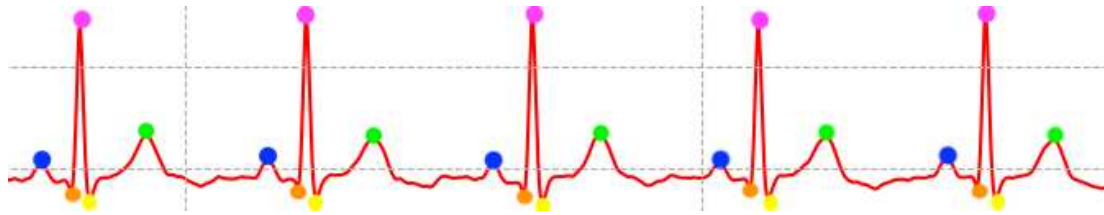
Prekordijalne elektrode stvaraju ostalih šest kanala. Kanali su istoimeni s nazivima odgovarajućih elektroda. Ti kanali promatraju električnu aktivnost srca na horizontalnom presjeku. Svih 10 elektroda zajedno tvori dvanaest kanala s kojima se dobiva potpuni prikaz električne aktivnosti srca. Slika 2.4 prikazuje 12 kanala stvorenih pomoću 10 elektroda.



Slika 2.4 – svih 12 kanala EKG-a [5]

2.2.2. Analiza snimke

Snimka EKG-a sastoji se od raznih valova. Svaki od tih valova predstavlja dio procesa rada srca. Na slici 2.6 prikazana je označena snimka EKG-a osobe s normalnim sinusnim ritmom. Plave, narandžaste, roze, žute i zelene točke redom odgovaraju vrhovima P valova, Q valova, R valova, S valova i T valova. Slike 2.5 – 2.16 iscrtane su u sklopu programske izvedbe zadatka.



Slika 2.5 – označena snimka normalnog sinus ritma (kanal I)

P val obično je prvi val koji se pronalazi gledajući snimku EKG-a. On predstavlja depolarizaciju oba atrija. Ovaj val započinje kada S-A čvor *okine*. P val normalno traje od 0.08 do 0.11 sekundi. Nakon P vala dolazi Tp val koji predstavlja repolarizaciju atrija. Tp val je obrnuto orijentiran od P vala. Budući da se odvija u isto vrijeme kada i QRS kompleks često ga ne vidimo na snimci. Dio snimke između kraja P vala i početka QRS kompleksa naziva se PR odsječkom. PR odsječak predstavlja prijenos električnog impulsa depolarizacije kroz čvor A-V, Hissov snop te Purkinjeova vlakna. PR interval predstavlja vremenski period od početka P vala do početka QRS kompleksa. On predstavlja faze rada srca obuhvaćene P valom i PR odsječkom. Normalno trajanje PR intervala iznosi između 0.11 i 0.20 sekundi.

QRS kompleks dolazi nakon P vala, tj. nakon PR segmenta. Sastoji se od dvaju ili više valova, uobičajeno od Q, R i S vala. QRS kompleks predstavlja depolarizaciju ventrikula. Q val prvi je negativni val QRS kompleksa. R val prvi je pozitivni val nakon P vala. S val dolazi nakon i on je negativan. QRS kompleks normalno traje od 0.06 do 0.11 sekundi. Od kraja QRS kompleksa do početka T vala proteže se ST odsječak. On predstavlja period mirovanja električnog naboja između depolarizacije i

repolarizacije ventrikula. Točka na snimci EKG-a gdje QRS kompleks završava i ST odsječak započinje naziva se točka J.

T val pozitivno je orijentirani val koji dolazi nakon QRS kompleksa, tj. netom nakon ST odsječka . On predstavlja repolarizaciju ventrikula. Vremenski period od početka QRS kompleksa do kraja T vala naziva se QT intervalom. Unutar QT intervala odvijaju se svi događaji sistole ventrikula. T valom završava jedan ciklus i nakon njega dolazi novi ciklus s P valom na čelu. Ipak, između T vala i P vala ponekad se može vidjeti malen i dosta spljošten val. Taj se val naziva U valom i ima istu orijentaciju kao i njegov prethodnik, T val.

2.3. Ritmovi i aritmije

U ovom poglavlju opisani su svi ritmovi, tj. većinom aritmije koje su prisutne u skupu podataka korištenom u ovom završnom radu. Opisan je normalni sinus ritam te osam različitih aritmija. Sve referencirane snimke su iz korištenog skupa podataka. Iscrtane su unutar implementacije opisane u 4. poglavlju.

Normalni sinusni ritam predstavlja normalno stanje rada srca gdje je čvor S-A glavni pejsmejker. Vremenski intervali na snimci, PR interval, QT interval, R-R interval, su ravnomjerni i unutar normalnih granica. Na slici 2.6 prikazana je snimka normalnog sinusnog ritma. Frekvencija rada srca iznosi između 60 i 100 otkucaja u minuti. Ako frekvencija iznosi više od 100 otkucaja/min radi se o sinus bradikardiji. Ako frekvencija iznosi manje od 60 otkucaja/min radi se pak o sinusnoj tahikardiji.



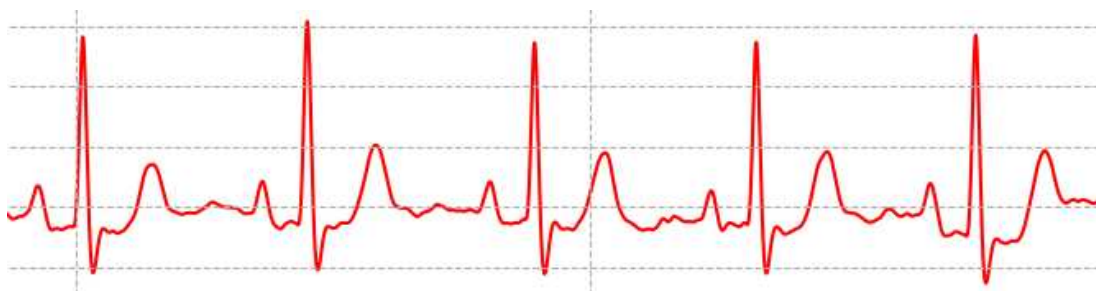
Slika 2.6 – snimka normalnog sinus ritma (kanal I)

Fibrilacija atriya brzi je i nepravilno nepravilan atrijski ritam. Obilježava ga kaotično okidanje raznih pejsmejкера u srcu. Pri fibrilaciji atriya sistola atriya nije prisutna, čvor A-V je zasut električnim impulsima. To uzrokuje neusklađeni prijenos impulsa i nepravilno nepravilnu sistolu ventrikula. Na snimci EKG-a se očituje odsutnošću jasnog P vala te nepravilno raspoređenim QRS kompleksima. Frekvencija otkucaja srca je obično u istom rasponu kao i tahikardija. Na slici 2.7 očituju se sve opisane značajke.



Slika 2.7 – snimka fibrilacije atriya (kanal I)

Atrioventrikulski blok (*A-V blok*) djelomičan je ili potpuni prekid provođenja električnih impulsa iz atriya u ventrikule. A-V blok prvog stupnja, aritmija koja je prisutna u korištenom skupu podataka, asimptomatska je aritmija te ne zahtjeva liječenje. Na snimci EKG-a, na slici 2.8 vidi se snimka, koja se očituje se produljenim PR intervalom, trajanja duljeg od 0.20 sekundi.



Slika 2.8 – snimka atrioventrikulskog bloka (kanal I)

Blok desne grane snopa aritmija je pri kojoj se električni impuls uspješno provodi kroz lijevu granu Hissovog snopa do lijevog ventrikula, a prijenos impulsa prema desnom ventrikulu je usporen. To uzrokuje sporiju depolarizaciju što se na snimci EKG-a očituje produljenjem QRS intervala na 0.12 sekundi ili više. Produljeni i spori S valovi još jedna su značajka. Na snimci prekordijalnog kanala V1 uočljiva je

dodatna značajka ove aritmije. QRS kompleksa izobličen je u takozvani oblik *zečjih ušiju* kao što je prikazano na slici 2.10.



Slika 2.9 – snimka bloka desne grane snopa (kanal I)

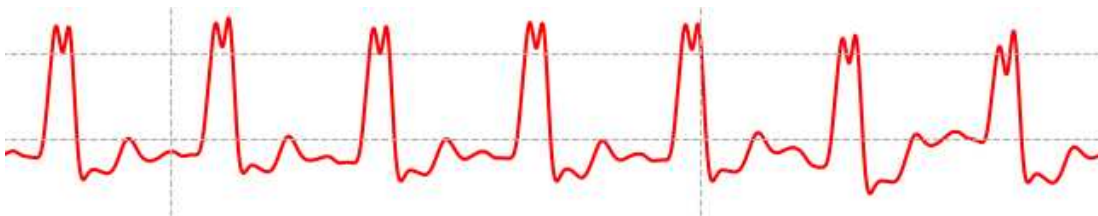


Slika 2.10 – snimka bloka desne grane snopa (kanal V1)

Blok lijeve grane snopa, obrnuto od bloka desne grane snopa, aritmija je pri kojoj električni impuls prvo prolazi desnom granom Hissovog snopa. Depolarizacija desnog ventrikula prethodi depolarizaciji lijevog. Na snimci, slika 2.11, očituju se široki R valovi bez prethodećih Q valova. Također, na snimci kanala V6 vidljiva je zanimljiva značajka, vrh R vala izgleda kao slovo *M*. Ta se značajka vidi na slici 2.12.

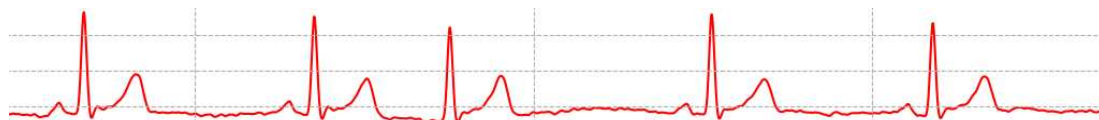


Slika 2.11 – snimka bloka lijeve grane snopa (kanal I)



Slika 2.12 – snimka bloka lijeve grane snopa (kanal V6)

Preuranjena kontrakcija atrijske aritmije je pri kojoj stanica s ulogom pejsmejke u atriju *okida* većom frekvencijom nego čvor S-A. To rezultira uranjenom pojavom PQRS ciklusa u jednom trenutku. Nakon toga se čvor S-A resetira, ali pauza zbog uranjenje kontrakcije atrijske nije nadomještena. Na slici 2.13 očigledna je preuranjena kontrakcija atrijske.



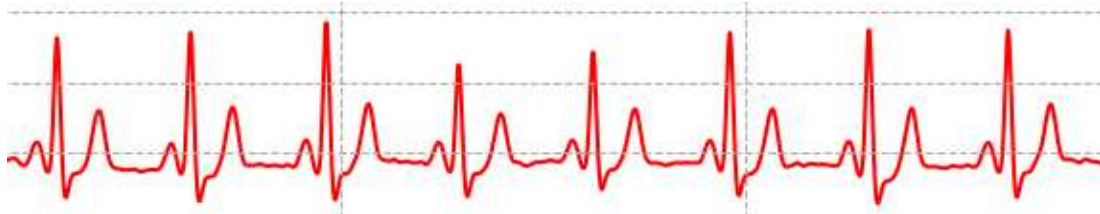
Slika 2.13 – snimka preuranjene kontrakcije atrijske (kanal I)

Preuranjena kontrakcija ventrikula, kao i već opisana preuranjena kontrakcija atrijske, događa se zbog bržeg *okidanja* stanice u ventrikulu nego čvora S-A. To uzrokuje uranjenu depolarizaciju ventrikula što ih stavlja u stanje oporavka srca. Kada normalni pejsmejker, čvor S-A, *okine*, ventrikuli nisu još repolarizirani. Zbog toga sistola ventrikula se ne odvija kada je predviđeno. Srce taj događaj ispravlja pauzom za kompenzaciju koja se na snimci EKG-a očituje vremenskim intervalom između dva vrha R valova između kojih se nalazi QRS kompleks bizarnog oblika.



Slika 2.14 – snimka preuranjene kontrakcije ventrikula (kanal I)

Promatrajući snimke EKG-a može se uočiti da odsječak ST, osim svog normalnog oblika, može biti izdignut ili potisnut. Ključno kod promatranja izdignutog ili potisnutog odsječka ST je uočavanje oblika. Oblik odsječka ST u pitanju može dati naznaku da se radi o ranoj repolarizaciji ventrikula (*perikarditis*), ozljedi srčanog mišića, deformaciji srčanog mišića, srčanoj ishemiji, ventrikularnoj aneurizmi ili pak nečemu drugom.



Slika 2.15 – potisnuti odsječak ST (kanal I)



Slika 2.16 – izdignuti odsječak ST (kanal V1)

3. Algoritmi

U ovom su poglavlju opisani algoritmi korišteni u implementaciji. U sljedećem poglavlju prikazane su njihove implementacije u radu. Algoritmi koji su opisani služe za analizu signala snimke EKG-a te za klasifikaciju strojnim učenjem. Uz razumijevanje domenskog znanja iz kardiologije, shvaćanje ovih algoritama također je bilo ključno u izradi ovog rada.

3.1. Analiza signala snimke EKG-a

Algoritmi za analizu signala snimke EKG-a bitan su korak u ovom završnom radu. Snimka EKG-a zapisana je kao niz signala. Pomoću ovih algoritama se iz snimke signala EKG-a izvlače ključne točke i informacije o samoj snimci. Time se pripremaju podaci za algoritme strojnog učenja. Ipak, analiza stvarne EKG snimke nije jednostavna. Ne samo zbog varijabilnosti u radu srca od pacijenta do pacijenta, već i zbog raznih vrsta mogućih šumova i smetnji prisutnih na snimci. Izvor tih šumova i smetnji najčešće su ostali mišići tijela, pomicanje elektroda i interferencija drugih izvora električne energije. Lutanje bazne linije na snimci i visoki T valovi koji liče na QRS kompleks također su smetnje pri analizi signala snimke EKG-a. U ovom poglavlju objašnjeniji su algoritmi za detekciju QRS kompleksa, P valova te T valova. Slike koje prikazuju rad ovih algoritama nalaze se u sljedećem poglavlju, *Implementacija*.

3.1.1. Pan Tompkinsov algoritam

Jiapu Pan i Willis J. Tompkins autori su znanstvenog rada pod nazivom “*A Real-Time QRS Detection Algorithm*” iz 1985. godine [6]. U radu je opisan algoritam za pouzdanu detekciju QRS kompleksa iz EKG signala u stvarnom vremenu. Izvorno je ovaj algoritam napisan u strojnom jeziku i implementiran na mikroprocesoru.

Algoritam koristi tri linearne transformacije signala te jednu nelinearnu kako bi što bolje riješio problem šuma u signalima. Obrada signala sastoji se od filtriranja pojasnim filterom, diferenciranja signala, uprosječivanja signala integratorom pomičnog prozora te kvadriranjem signala.

Pojasni filter smanjuje utjecaj šuma od strane mišića, interferencije T valova i lutajuće bazne linije. Propusni pojas poželjno je veličine 5 - 15 Hz. Izvorno je pojasni filter implementiran kaskadom niskopropusnog i visokopropusnog filtera kako bi se dobio propusni pojas veličine oko 5 - 12 Hz. Nakon toga signal se diferencira kako bi se dobila informacija o nagibu QRS kompleksa. Tada se signal kvadrira što signal pretvara u potpuno pozitivan signal te naglašava vrhove. Zadnji korak obrade signala je integracija pomičnim prozorom. Svrha toga je dobivanje informacija o obliku valova te detekcija vrhova R valova. Širina prozora bi trebala biti otprilike široka kao i najširi QRS kompleks na snimci. Ako je prozor preuzak, za neke QRS komplekse će se prikazati više vrhova nego je potrebno. Ako je prozor preširok, oblik koji integracija provede će obuhvaćati i T valove u QRS komplekse.

Iz prikaza generiranog integracijom pomičnim prozorom očituju se oblici specifični za taj prikaz. Rastući brid tog oblika odgovara QRS kompleksu, tj. trajanje rastućeg brida oblika jednako je širini QRS kompleksa. Vrh tog oblika odgovara vrhu R vala. Sljedeći korak algoritma je postavljanje granica kako bi se što točnije detektirali QRS kompleksi na snimci.

Određuju se dva skupa granica, jedan koji će se primjeniti na snimci generiranoj integracijom pomičnim prozorom, drugi koji će se primjeniti na snimci filtriranoj pojasnim filterom. Svaki skup sastoji se od sljedeće 4 granice: granica za vrh signala, granica za vrh šuma, viša granica te niža granica. Viša granica određena je vrijednostima granice za vrh signala i granice za vrh šuma. Kada algoritam naiđe na vrh, točka koja ima veću vrijednost od prethodne i nadolazeće točke uspoređuje njegovu vrijednost sa snimke sa višom granicom. Ako je vrijednost vrha veća od više granice, točka se klasificira kao vrh. To potiče preračunavanje granice za vrh signala. Ako unutar određenog vremenskog intervala nije klasificiran niti jedan vrh, taj se

interval ponovno pretražuje, no ovaj se put koristi niža granica koja je upola manja od više granice. Kako bi se identificirao QRS kompleks, njegov vrh mora biti klasificiran u obje snimke. Vrlo važno je vremensko trajanje oporavka srca, točnije 200 ms. Dakle, vremenska udaljenost dvaju R vrhova mora biti barem veća od 200 ms.

Pored svega navedenoga, algoritam također održava dvije prosječne vrijednosti R-R intervala. Jedna je prosječna vrijednost osam zadnje viđenih R-R intervala, otkucaja. Druga je prosječna vrijednost osam zadnje viđenih R-R intervala koji pripadaju određenom vrijednosnom intervalu. Razlog ovoj tehnici je mogućnost brzog prilagođavanja nepravilnim frekvencijama otkucaja srca. Ukoliko su za zadnjih osam otkucaja obje prosječne vrijednosti R-R intervala iste, radi se o pravilnom srčanom ritmu.

Također, algoritam ima sigurnosnu provjeru kod klasifikacije QRS kompleksa. Ukoliko je R-R interval kraći od 360 ms, ali naravno dulji od 200 ms, procjenjuje se radi li se o T valu ili o QRS kompleksu. Ukoliko je najveći nagib oblika u prikazu signala integracijom pomičnog prozora manji od polovice prijašnjeg, oblik se klasificira kao T val. U suprotnom klasificira se kao QRS kompleks.

3.1.2. Elgendijev algoritam

Mohamed Elgendi, Bjoern Eskofier i Derek Abbot autori su znanstvenog rada "*Fast T Wave Detection Calibrated by Clinical Knowledge with Annotation of P and T Waves*" iz 2015. godine [7]. Kao što se može shvatiti iz naslova rada, ovaj algoritam služi za detekciju T valova. Algoritam se sastoji od tri glavne faze: obrada signala, izlučivanje značajki i klasifikacija.

Obrada signala prva je faza i ključna je za dobivanje snimke signala s kojom algoritam može raditi. Prvi dio obrade je filtriranje pomoću Butterworthovog filtera drugog reda s propusnim pojasom veličine 0.5 - 10 Hz. To filtriranje smanjuje utjecaj

lutanja bazne linije i visokih frekvencija koje ne doprinose T valovima. Sljedeći korak u obradi signala je uklanjanje QRS kompleksa. Transformacijom signala u širini QRS kompleksa u signale vrijednosti 0, T valovi postaju istaknutiji na snimci. Također, poznavanjem lokacije vrha R vala lakše je odrediti T valove u odnosu na udaljenost od vrha R vala.

Izlučivanje značajki u ovom algoritmu zasniva se na principu interesnih blokova (*engl. Blocks of interest*) te pokretnih prosjeka (*engl. Moving averages*). Generiraju se interesni blokovi koristeći dva pokretna prosjeka kako bi se označili T valovi na snimci. Prvi pokretni prosjek (MA_{peak}) stvara se prema formuli 3.1 i naglašava područje vrha T vala. Drugi pokretni prosjek (MA_{Twave}) stvara se prema formuli 3.2 i naglašava prostor T vala, tj. služi kako bi pri usporedbi s prvim pokretnim prosjekom označio T val. Oznaka W_1 predstavlja širinu prozora, otprilike široku kao trajanje vrha T vala. Oznaka W_2 također predstavlja širinu prozora, no širinu trajanja T vala. Nakon što su određeni pokretni prosjeci oni se slijedno uspoređuju. Interesni blokovi su područja na snimci signala EKG-a gdje prvi pokretni prosjek ima veću vrijednost nego drugi pokretni prosjek.

$$MA_{peak}[n] = \frac{1}{W_1} (y[n - (W_1 - 1)/2] + \dots + y[n] + \dots + y[n + (W_1 - 1)/2]) \quad (3.1)$$

$$MA_{Twave}[n] = \frac{1}{W_2} (y[n - (W_2 - 1)/2] + \dots + y[n] + \dots + y[n + (W_2 - 1)/2]) \quad (3.2)$$

Konačna faza ovog algoritma, klasifikacija, sastoji se od analize interesnih blokova dobivenih iz prethodne faze. U ovu fazu najčešće se dolazi s velikim brojem interesnih blokova, no većina njih zapravo su P valovi, U valovi ili šum. Prvo se određuje maksimalna i minimalna moguća udaljenost T vala od trenutnog R vala prema formulama 3.3 i 3.4. D_{min} iznosi 170 ms, a D_{max} iznosi 800 ms. Svi interesni blokovi koji zadovoljavaju maksimalnu i minimalnu granicu uspoređuju se s graničnom vrijednošću trajanja THR_1 koja je jednaka vrijednosti W_1 . Ukoliko je širina interesnog bloka veća ili jednaka graničnoj vrijednosti trajanja, interesni se blok klasificira kao T val. Zadnja točka ove faze je traženje vrijednosti vrhova T

valova. Te se vrijednosti pronalaze traženjem maksimalne apsolutne vrijednosti u svakom interesnom bloku koji je klasificiran kao T val.

$$R_i T_{min} = D_{min} R_i R_{i+1} \quad (3.3)$$

$$R_i T_{max} = D_{max} R_i R_{i+1} \quad (3.4)$$

3.2. Strojno učenje

Kako bi se riješio klasifikacijski problem ovog završnog rada potrebno je razumjeti i koristiti algoritme strojnog učenja. Točnije, ovdje se koriste i opisuju algoritmi stroja s potpornim vektorima te logističke regresije. Slike u ovom poglavlju vlastoručno su izrađene. Korišteni izvori za pisanje ovog poglavlja su [8],[9], [10] i [11].

3.2.1. Stroj s potpornim vektorima

Stroj s potpornim vektorima (*engl. Support Vector Machine*) linearni je model za podržano učenje kojim se rješava problem binarne klasifikacije. Stroj s potporim vektorima stvara hiperravninu koja razdvaja vektore, uzorke podataka, u razrede. Potporni vektori najbliži su vektori hiperravnini. Udaljenost hiperravnine od potpornih vektora naziva se marginom. Ovaj algoritam tu marginu nastoji maksimizirati. Time se postiže dobra generalizacija što je cilj svakog algoritma strojnog učenja. Ono što razlikuje algirtam stoja s poptornim vektorima od klasifikatora optimalne granice (*engl. Maximum Margin Classifier*) je jezgreni trik koji ovaj algoritam koristi. Kod stvaranja optimalne hiperravnine za klasifikaciju podaci mogu biti linearno razdvojivi ili ne. Ukoliko nisu linearno razdvojivi, koristi se metoda meke granice (*engl. Soft margin*) koja dopušta pogrešnu klasifikaciju u svrhu bolje generalizacije.

Za dva razreda, dakle riječ je o binarnoj klasifikaciji, iskoristimo oznake -1 i $+1$. Uzorak iz podataka može se prikazati kao skup $X = \{x^t, y^t\}$ gdje je $y^t = +1$ ako $x^t \in C_1$ i $y^t = -1$ ako $x^t \in C_2$. Sada je bitno pronaći w , normalu hiperravnine, i w_0 takve da zadovoljavaju:

$$h(x^t) \geq +1 \text{ za } y^t = +1$$

$$h(x^t) \geq -1 \text{ za } y^t = -1$$

$$\text{gdje } h(x) = w^T x + w_0$$

Uz pretpostavku linearne razdvajivosti, kombinacijom ovih dvaju izraza dobivamo izraz koji mora biti zadovoljen:

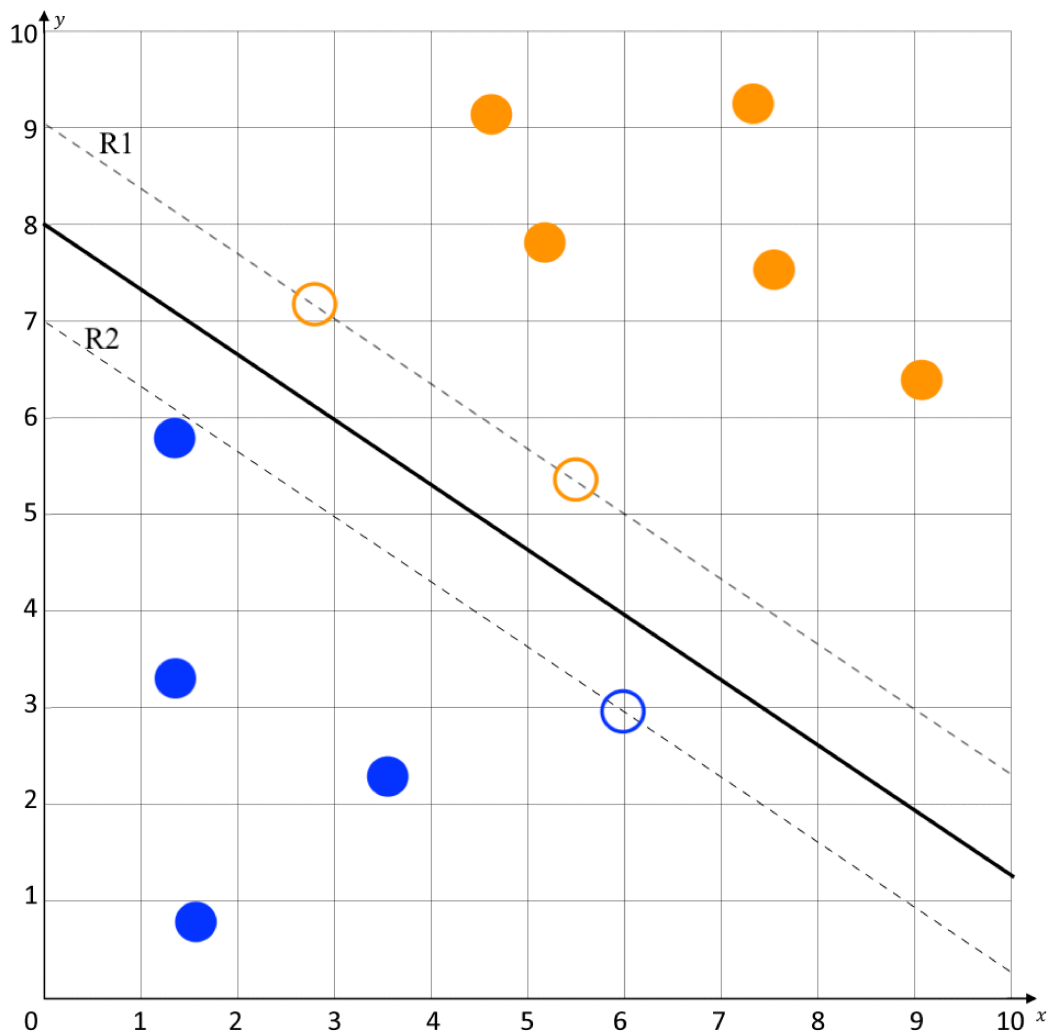
$$y^t h(x^t) \geq +1, \forall t \quad (3.5)$$

Na slici 3.6 vidi se prikaz ključnih dijelova algoritma, hiperravnina koja odvaja razrede, ravnine R1 i R2 na kojima leže potporni vektori označeni kao kružići te obični vektori označeni kao točke u različitim bojama predstavljajući svoj razred. Očito je da su ravnine R1 i R2 jednako udaljene od hiperravnine koja dijeli vektore u dva razreda. Za ravnine R1 i R2 vrijedi sljedeće:

$$h(x^t) = -1 \text{ za R1}$$

$$h(x^t) = +1 \text{ za R2}$$

Kako w označava normalnu hiperravnine, udaljenost ravnina R1 i R2 od hiperravnine može se definirati kao $\frac{1}{\|w\|}$. Ta vrijednost označava marginu stroja s potpornim vektorima, tj. to je vrijednost koju nastojimo maksimizirati. Maksimiziranje margine isto je što i traženje minimalne vrijednosti $\|w\|$ tako da vrijedi izraz 3.5.



Slika 3.6 – stroj s potporim vektorima u dvodimenzionalnom koordinatnom sustavu

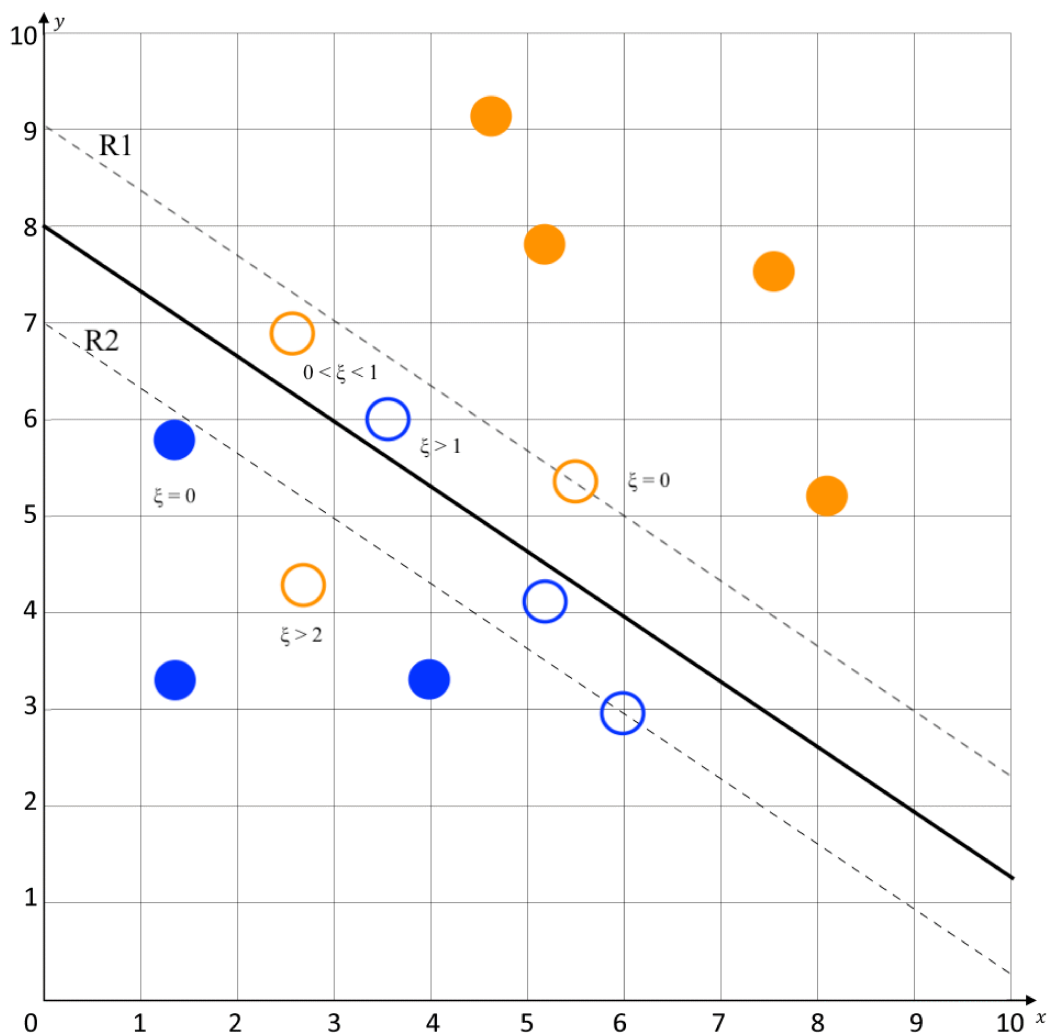
Kada podaci nisu linearno razdvojivi opisani algoritam neće funkcionirati jer ne postoji hiperravnina koja je u stanju razdvojiti linearno nerazdvojive podatke. Zbog toga tražimo onu hiperravninu koja će klasificirati podatke uz što manju grešku. U tu svrhu uvode se vrijednosti $\xi^t \geq 0$ koje govore koliko je uzorak x^t ušao u marginu $\xi^t = |y^t - h(x^t)|$. Ako je $\xi^t = 0$, uzorak je točno klasificiran. Ako je $0 \leq \xi^t \leq 1$, uzorak je ispravno klasificiran, ali je ušao u marginu. Ako je pak $\xi^t > 1$, uzorak je pogrešno klasificiran. Uvođenjem ove vrijednosti ublažujemo uvjete izraza 3.5 na:

$$y^t h(x^t) \geq 1 - \xi^t, \forall t$$

Hiperravnine se u ovom slučaju pronalazi rješavanjem:

$$\min \frac{1}{2} \|w^2\| + C \sum_{t=1}^N \xi^t \text{ gdje je } N \text{ broj uzoraka i vrijedi } y^t h(x^t) \geq 1 - \xi^t, \forall t$$

Opisana metoda naziva se metodom meke margine, a prikazana je na slici 3.7. Vrijednost C označava faktor pogreške bez kojeg pronalazak hiperravnine za linearno nerazdvojive podatke ne bi bio moguć.



Slika 3.7 – metoda meke margine

Umjesto korištenja nelinearnog modela kada su podaci nelinearni, moguće je preslikavanje svih vektora iz n -dimenzionalnog prostora u k -dimenzionalni prostor koristeći baznu funkciju $\phi(x^t)$. U k -dimenzionalnom prostoru može se koristiti linearni model za rješavanje problema, a najčešće je k znatno veći od n . No sada

dolazi do novog problema, a to je računanje skalarnog produkta vektora u novom prostoru, koji može biti i beskonačan. U svrhu rješavanja tog problema stroj s potpornim vektorima koristi se *jezgrenim trikom* (engl. *kernel trick*). Što zapravo radi je to da skalarni produkt vektora zamjenjuje s jezgrenom funkcijom koju je najčešće moguće mnogo jednostavnije izračunati:

$$\kappa(x, x') = \phi(x)^T \phi(x')$$

Najčešće korištene jezgrene funkcije su:

- Linearna:

$$\kappa(x, x') = x^T x'$$

- polinomijalna stupnja d :

$$\kappa(x, x') = (\gamma x^T x' + c)^d$$

- gaussova radijalna bazna funkcija:

$$\kappa(x, x') = \exp(-\gamma \|x - x'\|)$$

Iako je stroj s potpornim vektorima binarni klasifikator, moguće je provesti i klasifikaciju između više razreda. Ipak takvu klasifikaciju nije moguće provesti samo jednim klasifikatorom već izgradnjom više binarnih klasifikatora. Jedna od ovdje korištenih metoda je *jedan-protiv-jednog* (engl. *one-versus-one*, *OvO*). Počinje se izgradnjom $\frac{N(N-1)}{2}$ binarnih klasifikatora, gdje N predstavlja broj razred. Zapravo, ova metoda od početnog klasifikacijskog problema stvara puno binarnih klasifikacijskih problema, takvih da pokrije sve moguće parove razreda u izvornom problemu. Ako klasifikacijski problem ima tri moguća razreda x , y i z , ova će metoda stvoriti tri binarna klasifikacijska problema, x ili y , x ili z i y ili z . Za svaki binarni klasifikator korišten je novi skup podataka koji je zapravo odrezak izvornog skupa podataka tako da odgovara određenom problemu. Pri klasifikaciji odluka o konačnoj klasifikaciji dolazi iz glasovanja svih $\frac{N(N-1)}{2}$ binarnih klasifikatora. Svaki klasifikator

glasuje za razred koji je klasificirao za uzorak. Na kraju, razred s najviše glasova odabran je kao konačni razred izvornog klasifikacijskog problema za taj uzorak.

3.2.2. Logistička regresija

Logistička regresija linearni je model koji se koristi za klasifikaciju. Primjer je poopćenog linearnog modela (engl. *generalized linear model, glm*). U poopćenim linearnim modelima neka je nelinearna funkcija f omotana oko skalarnog produkta vektora težine i vektora značajki. Takva funkcija f naziva se aktivacijskom funkcijom i ona preslikava skalarni produkt u ograničeni interval, primjerice u $[0, 1]$ ili $(-1, +1)$. Aktivacijska funkcija logističke regresije je logistička funkcija (sigmoida):

$$\sigma(\alpha) = \frac{1}{1 + \exp(-\alpha)}$$

Sigmoida, prikazana na slici 3.8, ima tri važne karakteristike koje ju čine dobrim odabirom. Ona ograničava vrijednost na interval $[0, 1]$, što znači da se izlazne vrijednosti mogu interpretirati kao vjerojatnosti. Dobro klasificira u smislu da će primjerima iz jednog razreda dodijeliti vrijednosti blizu 1, a primjerima iz drugog razreda blizu 0. Također sigmoida, je derivabilna što omogućuje izvođenje učinkovitog postupka optimizacije.

Dakle, model logističke regresije je, slika 3.8:

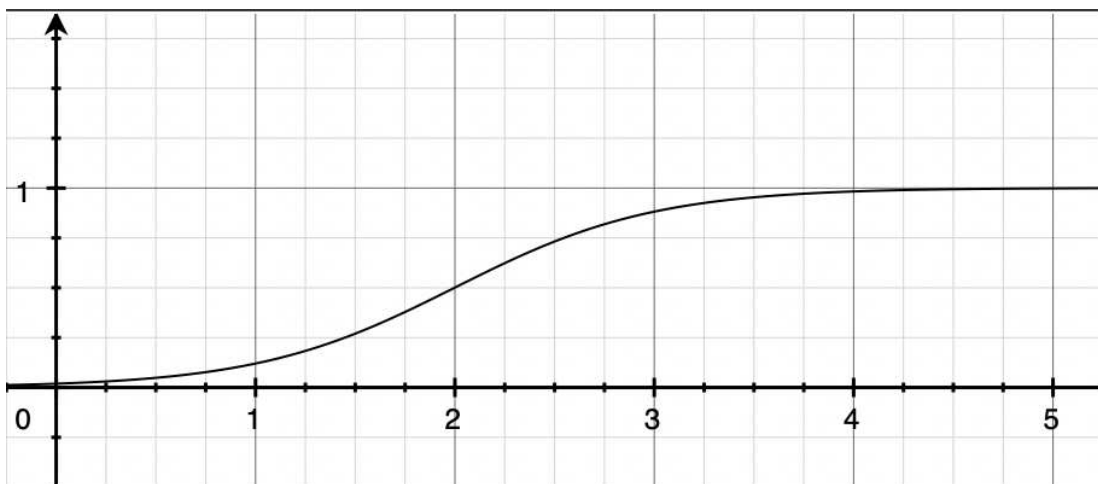
$$h(x; w) = f(w^T \phi(x)) = \sigma(w^T \phi(x)) = \frac{1}{1 + \exp(-w^T \phi(x))}$$

Konačni izraz zapravo je ekvivalentan vjerojatnosti $P(y = 1|x)$. Može se zaključiti da se radi o binarnom klasifikacijskom modelu. Također, iz intervala $[0, 1]$, na koji aktivacijska funkcija ograničava izlaz, model može se zaključiti da je granica između razreda definirana hiperravninom u prostoru značajki. Za tu hiperravninu vrijedi $h(x; w) = 0.5$. Zbog te značajke model možemo definirati i na sljedeći način:

$$h(x; w) = 1 \text{ za } \sigma(w^T \phi(x)) \geq 0.5$$

$$h(x; w) = 0 \text{ inače}$$

Međutim, takva definicija nije poželjna jer se njenim korištenjem gubi informacija o pouzdanosti klasifikacije koju se zaključuje iz vrijednost sigmoide.



Slika 3.8 – grafički prikaz logističke funkcije

4. Implementacija

U ovom poglavlju opisana je programska implementacija zadatka ovog završnog rada. Prvo je opisan ulazni skup podataka, zatim programska izvedba i korištenje algoritama opisanih u prethodnom poglavlju. Naposljetku su opisani dobiveni rezultati ove implementacije za problem klasifikacije srčanih ritmova iz snimaka 12-kanalnog elektrokardiograma.

4.1. Podaci

Ovaj završni rad koristi javno dostupni skup podataka [12] od međunarodnog izazova CinC 2020 [13]. Taj skup podataka izvorno je korišten na izazovu CPSC2018 [14]. Snimke EKG-a sakupljene su iz 11 bolnica. Korišteni skup podataka sadrži 6877 snimaka 12-kanalnog EKG-a u trajanju od 6 do 60 sekundi. Svaki uzorak skupa podataka sastoji se od binarne datoteke *MATLAB v4 (.mat)* i tekstualne datoteke u obliku *WFDB header (.hea)*. Binarne datoteke *.mat* sadrže snimku signala EKG-a, dok tekstualne datoteke *.hea* sadrže opis snimke signala i značajke pacijenta. Na slici 4.1 prikazan je primjer zapisa iz datoteke *A0010.hea*:

```
A0010 12 500 5000 05-Feb-2020 11:39:16
A0010.mat 16+24 1000/mV 16 0 159 58 0 I
A0010.mat 16+24 1000/mV 16 0 68 124 0 II
A0010.mat 16+24 1000/mV 16 0 -91 19 0 III
A0010.mat 16+24 1000/mV 16 0 -114 -71 0 aVR
A0010.mat 16+24 1000/mV 16 0 125 -2 0 aVL
A0010.mat 16+24 1000/mV 16 0 -11 58 0 aVF
A0010.mat 16+24 1000/mV 16 0 -267 -151 0 V1
A0010.mat 16+24 1000/mV 16 0 -192 -61 0 V2
A0010.mat 16+24 1000/mV 16 0 -123 3 0 V3
A0010.mat 16+24 1000/mV 16 0 -103 44 0 V4
A0010.mat 16+24 1000/mV 16 0 -25 95 0 V5
```

```
A0010.mat 16+24 1000/mV 16 0 40 117 0 V6
#Age: 82
#Sex: Male
#Dx: RBBB
#Rx: Unknown
#Hx: Unknown
#Sx: Unknowns
```

Slika 4.1– sadržaj datoteke A0010.heg

Iz prvog retka zapisa očituje se sljedeće. Identifikacijska oznaka snimke je *A0010*, dakle datoteka *A0010.mat* odgovara ovoj datoteci *.heg*. Snimka je snimljena na EKG-u s 12 kanala. Frekvencija uzorkovanja iznosi 500 Hz, a snimka sadži 5000 uzoraka, tj. signala. Naposljetku je zapisana vremenska oznaka snimanja.

Sljedećih 12 redaka odgovara svakom od 12 kanala EKG-a, a na kraju svakog retka zapisan je kanal kojem redak odgovara. Redak počinje nazivom odgovarajuće datoteke *.mat*. Zatim slijedi informacija o zapisu signala, gdje je vidljivo da je ovaj signal zapisan s pomoću 16 bitova s posmakom od 24 bita. Amplituda razlučivosti snimke iznosi 1000 jedinica po milivoltu (*mV*). Razlučivost analogno-digitalnog pretvarača iznosi 16 bita. Vrijednost stvarnog signala od 0 predstavljena je u zapisu kao vrijednost 0. Zatim je zapisana prva vrijednost signala, u slučaju kanala *V4* iznosi -103. Nakon prve vrijednosti signala slijedi kontrolni zbroj, u slučaju kanala *aVR* iznosi -71.

Iz zadnja tri retka datoteke *.heg* isčitava se sljedeće. Pacijent je 82-godišnji muškarac, što se vidi iz redaka *Age* i *Sex*. Dijagnosticiran je s blokom desne grane snopa (RBBB), što se vidi iz retka *Dx*. Za ovog pacijenta liječnički recept (*Rx*), povijest bolesti (*Hx*) i simptomi kirurškog zahvata (*Sx*) su nepoznati.

Ciljna oznaka u ovom klasifikacijskom problemu je dijagnoza pacijenta. Vrijednosti koje ta oznaka može sadržavati su sljedeće:

1. Normal – normalni sinusni ritam,
2. AF – fibrilacija atrija,
3. I-AVB – atrioventrikulski blok prvog stupnja,
4. LBBB – blok lijeve grane snopa,
5. PAC – preuranjena kontrakcija atrija,
6. PVC – preuranjena kontrakcija ventrikula,
7. RBBB – blok desne grane snopa,
8. STD – potisnuti odsječak ST,
9. STE – izdignuti odsječak ST,

4.2. Programska izvedba

Programska izvedba ovog završnog rada strukturirana je na sljedeći način:

- main.py
- ecg12_processing
 - morphological_analysis.py
 - fiducial_points_detection
 - Butterworth.py
 - pt_detection
 - Elgendi.py
 - qrs_detection
 - Pan_Tompkins.py

Programski kod pisan je u programskom jeziku *python* [15]. Vanjski moduli *scipy* [17] te *numpy* [16] standardni su moduli za korištenje u području podatkovne znanosti. Uz njih, korišten je i vanjski modul *pandas* [18] za zapis i čitanje *csv* datoteka. Za prikazivanje snimaka EKG-a korišten je vanjski modul *matplotlib* [19]. Naposljetku, za implementaciju algoritama strojnog učenja korišten je vanjski modul *scikit-learn* [20].

4.2.1. Tok programa

Podaci su čitani iz datoteka *.mat* koristeći funkciju *loadmat* iz modula *io* vanjskog modula *scipy* [17]. Podaci se čitaju pojedinačno, za svaki uzorak prvo se čita zaglavlje. Ukoliko je redak *Age* nevaljan ili je redak *Dx* kombinacija više dijagnoza, uzorak se odbacuje i nastavlja se na čitanje novoga. Uzorci podataka koji nisu odbačeni predaju se funkciji *detect_qrs* modula *Pan_Tompkins*. Povratna vrijednost iz te funkcije, podaci o QRS kompleksima na snimci, sprema se u varijablu *qrs_complexes* koja se zatim predaje funkciji *detect_p_t* modula *Elgendi*. Povratna vrijednost te funkcije, podaci o P i T valovima na snimci, sprema se u varijablu *p_t_waves* koja se uz varijablu *qrs_complexes* predaje funkciji *analyze* modula *morphological_analysis*. Ta funkcija vraća morfološke značajke snimke EKG-a u obliku riječnika. Tok programa do ovog trenutka prikazan je na isječku koda 4.2. Konačno, te morfološke značajke, uz spol, dob i dijagnozu pacijenta spremaju se i konačno zapisuju u datoteku *.csv*.

```
ecg_data = challenge_data['data'][0]
qrs_complexes = Pan_Tompkins.detect_qrs(ecg_data, fs, plot=False)
p_t_waves = Elgendi.detect_p_t(ecg_data, fs, qrs_complexes,
plot=False)
Features = morphological_analysis.analyze(ecg_data, qrs_complexes,
p_t_waves, fs, plot=False)
```

Isječak koda 4.2 – obrada podataka

Transformacijom izvornog skupa podataka dobio se novi skup podataka koji je spreman za korištenje u algoritmu strojnog učenja. Iz modula *scikit-learn* [20] korišteni su gotovi algoritmi za logističku regresiju i stroj s potpornim vektorima. Za validaciju algoritama na ovom problemu koristile su se mjere točnosti klasifikacije i F1 mjera, a optimizacija parametara provela se pomoću unakrsne validacije s *k* preklopa.

4.2.2. Izvedba Pan-Tompkinsovog algoritma

Izvedba Pan-Tompkinsovog algoritma nalazi se u modulu *Pan_Tompkins.py*, točnije u funkciji *detect_qrs*. Funkcija od parametara prima snimku signala EKG-a, frekvenciju uzorkovanja i zastavicu za grafički prikaz podataka čija je pretpostavljena vrijednost *False*.

```
detect_qrs(ecg, fs, plot=False)
```

Pojasni filter ostvaruje se korištenjem Butterworthovog pojasnog filtera prvog reda pojasa 5-15 Hz. Poziva se funkcija *bandpass_filter*, prikazano u isječku koda 4.3, implementirana u modulu *Butterworth.py*, implementacija je prikazana u isječku koda 4.4.

```
from ..Butterworth import bandpass_filter
ecg_filtered = bandpass_filter(ecg, filter_lowcut,
                              filter_highcut, fs, filter_order)
```

Isječak koda 4.3 – poziv metode *bandpass_filter*

```
from scipy.signal import butter, lfilter
def bandpass_filter(data, lowcut, highcut, fs, filter_order):
    nyquist_freq = 0.5 * fs
    low = lowcut / nyquist_freq
    high = highcut / nyquist_freq
    b, a = butter(filter_order, [low, high], btype="band")
    y = lfilter(b, a, data)
    return y
```

Isječak koda 4.4 – implementacija metode *bandpass_filter* iz modula *Butterworth.py*

Ostale faze obrade signala opisane u trećem poglavlju odvijaju se jednako, prikazano je u isječku koda 4.5. Svaki signal se naknadno i normalizira.

```
ecg_differentiated = np.ediff1d(ecg_filtered)
```

```

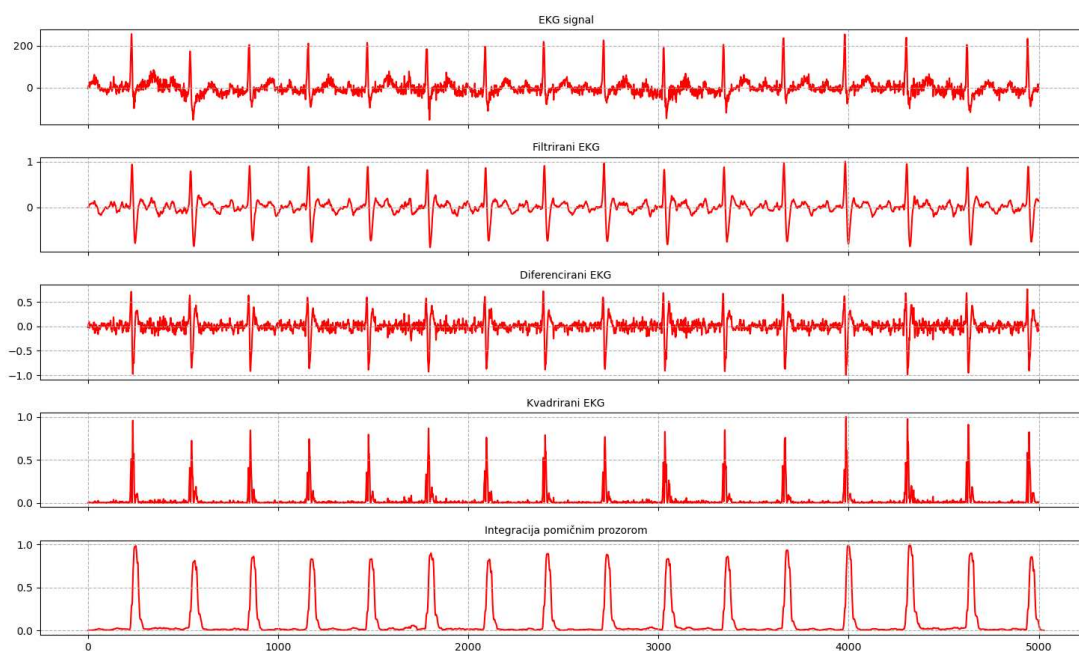
ecg_differentiated = ecg_differentiated / np.max(ecg_differentiated)

ecg_squared = ecg_differentiated ** 2
ecg_squared = ecg_squared / np.max(ecg_squared)

ecg_integrated = np.convolve(ecg_squared,
                              np.ones(integration_window) / integration_window)
ecg_integrated = ecg_integrated / np.max(ecg_integrated)

```

Isječak koda 4.5 – faze obrade signala nakon filtriranja pojasnim filterom



Slika 4.6 – transformacija izvornog signala u 4 koraka Pan-Tompkinsovog algoritma



Slika 4.7 – uklanjanje lutajuće bazne linije filtriranjem signala i ostala 3 koraka transformacije

Tada se poziva funkcija `find_peaks(ecg, spacing, peak_limit)` koja pronalazi vrhove oblika dobivenih integracijom pomičnim prozorom. Pronalazi ih tako da prolazi signalom `ecg` vrijednost po vrijednost. Kada pronade točku na signalu takvu da joj je vrijednost veća od prethodne i manja od sljedeće i veća od vrijednosti `peak_limit` kandidira ju za vrh. Tada se pregledavaju sljedećih vrhovi u udaljenosti od `spacing` od kandidirane točke. Ako se pronade točka koja zadovoljava iste uvjete, ali ima veću vrijednost, ona postaje kandidat. Prije nove iteracije trenutni kandidat se sprema u polje vrhova, a iterator se poveća za `spacing`. Funkcija vraća polje vrhova `peak_indices`. Ukoliko vraćeno polje nema dovoljno elemenata, ponovno se poziva funkcija ali ovoga puta s upola manjom vrijednosti `peak_limit`.

Sljedeći korak je odluka o svakom vrhu u polju vrhova. Stvara se dodatno polje koje sadrži vrijednosti vrhova iz polja vrhova. Inicijaliziraju se granice: granica vrha signala, granica vrha šuma i viša granica. Niža granica nije korištena jer se njoj posao većinom rješio u prethodnom koraku. Iterira se u isto vrijeme po obje liste s informacijama o vrhovima pomoću ugrađene funkcije `zip`. Za svaki vrh provjerava se je li se već neki vrh detektirao kao QRS kompleks, a ako jest, jeli njihov razmak veći od vremena oporavka srca. Ako su ovi uvjeti zadovoljeni ispituje se je li vrijednost vrha veća od gornje granice i time se određuje radi li se o QRS kompleksu ili o šumu. Naposljetku se označava početak, kraj i vrh QRS kompleksa koristeći polje detektiranih vrhova QRS kompleksa i signal dobiven integracijom pomičnog prozora. Implementacija ovog postupka prikazana je u isječku koda 4.8, a na slici 4.9 prikazan je rezultat algoritma, tj. demarkacija QRS kompleksa.

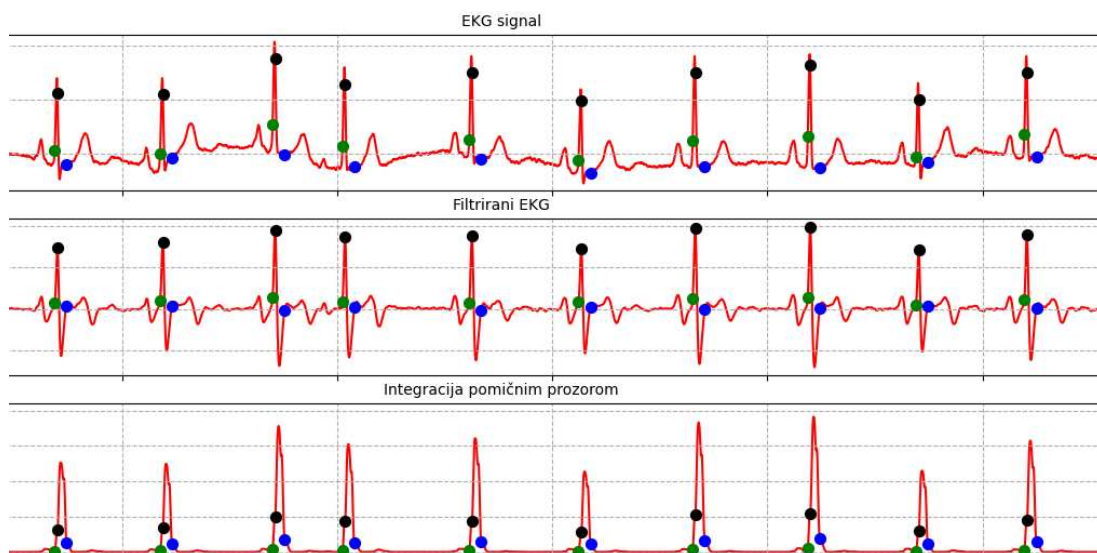
```
detected_qrs = np.array([], dtype=int)
detected_noise = np.array([], dtype=int)
for ind, value in zip(peak_indices, peak_values):
    if detected_qrs.size == 0 or ind - detected_qrs[-1] >
        refractory_period:
        if value > threshold_1:
            detected_qrs = np.append(detected_qrs, ind)
            signal_peak = 0.125 * value + 0.875 * _peak
        else:
            detected_noise = np.append(detected_noise, ind)
```

```

noise_peak = 0.125 * value + 0.875 * noise_peak
threshold_1 = noise_peak + 0.25 * (signal_peak - noise_peak)

```

Isječak koda 4.8 – izbor QRS kompleksa u Pan-Tompkinsovom algoritmu



Slika 4.9 – označene točke korištenjem Pan Tompkinsovog algoritma

4.2.3. Izvedba Elgendijevog algoritma

Izvedba Elgendijevog algoritma nalazi se u modulu *Elgendi.py*, točnije u funkciji *detect_p_t*. Pri pozivu funkcije predaju joj se sljedeći argumenti: snimka signala EKG-a, informacija o QRS kompleksima, frekvencija uzorkovanja i zastavica za grafički prikaz čija je pretpostavljena vrijednost *False*.

```
detect_p_t(ecg, qrs, fs, plot=False)
```

Filtriranje signala Butterworthovim pojasnim filterom drugog reda ostvaruje se pozivom funkcije *bandpass_filter*. Dobiveni signal će imati izražene P i T valove zbog ovakvog filtriranja. Sljedeći korak je uklanjanje QRS kompleksa što se izvodi postavljanjem vrijednosti signala od početka do kraja QRS kompleksa na nulu. Ovo se radi kako QRS kompleksi ne bi smetali u pronalaženju interesnih blokova. Nakon uklanjanja dolazi stvaranje pokretnih prosjeka. U isječku koda 4.10 prikazana je implementacija stvaranja pokretnih prosjeka.

```

w1 = 0.05 * fs
a = (w1 - 1) / 2
w2 = 0.12 * fs
b = (w2 - 1) / 2
ma_peak = np.ones(ecg_filtered.size)
ma_t_wave = np.ones(ecg_filtered.size)

for n in range(0, ecg_filtered.size):
    ma_peak[n] = np.round(np.sum(ecg_filtered[int(n -
                                                a):int(n + a)])) / w1)
    ma_t_wave[n] = np.round(np.sum(ecg_filtered[int(n -
                                                b):int(n + b)])) / w2)

```

Isječak koda 4.10 – stvaranje pokretnih prosjeka

Jednom kada su pokretni prosjeci *ma_peak* i *ma_t_wave* stvoreni, počinje generiranje interesnih blokova. Polje interesnih blokova inicijalizira se kao prazni *np.array*. Usporedba se radi za svaki element pokretnih prosjeka. Ako je element polja *ma_peak* veći od elementa polja *ma_t_wave*, u polje interesnih blokova na toj poziciji dodaje se 1, u protivnom dodaje se 0. U isječku koda 4.11 prikazana je implementacija opisanog.

```

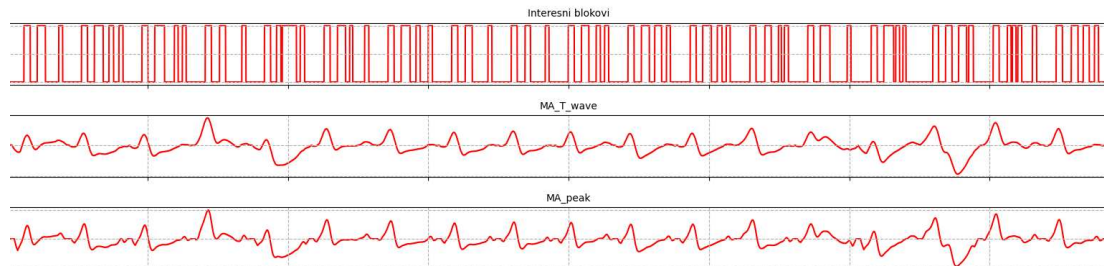
blocks = np.array([], dtype=int)
for peak, wave in zip(ma_peak, ma_t_wave):
    if peak > wave:
        blocks = np.append(blocks, 1)
    else:
        blocks = np.append(blocks, 0)

```

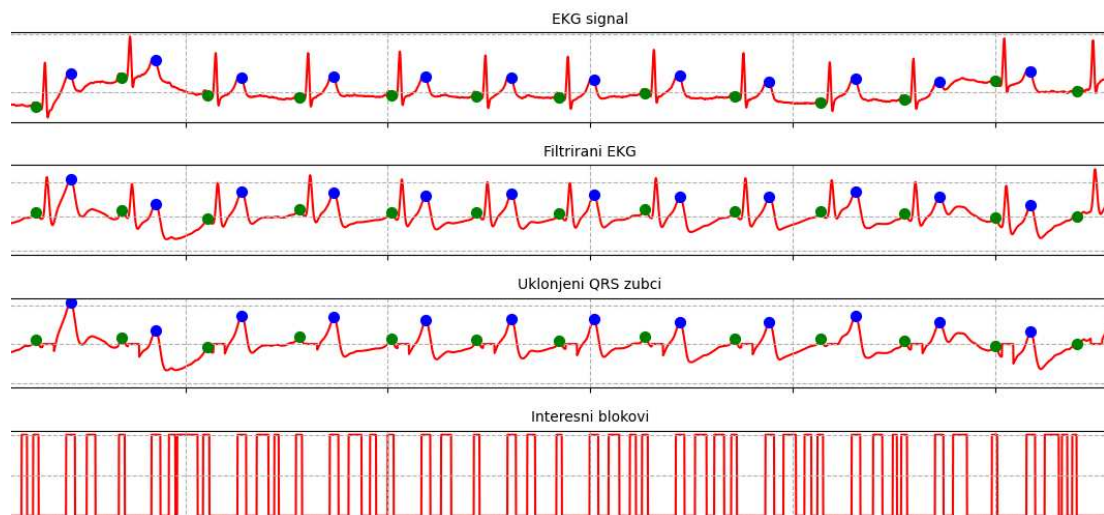
Isječak koda 4.11 – stvaranje interesnih blokova uspoređivanjem elemenata pokretnih prosjeka

Nakon toga, iz polja *blocks* dobiva se lista objekata razreda *Block*. Svrha ovog koraka je lakše snalaženje u interesnim blokovima. Sada kreće postupak odbacivanja blokova koji su ili preuski ili preblizu QRS kompleksima. Postupak je reguliran parametrima koji određuju minimalnu širinu interesnog bloka i minimalnu i maksimalnu udaljenost T i P vala od odgovarajućih vrhova QRS kompleksa.

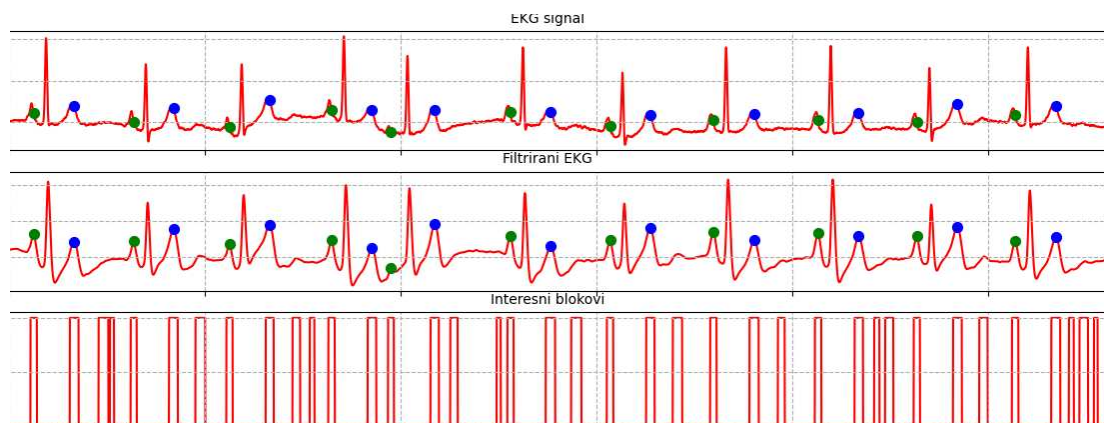
Konačno, kao povratna vrijednost, šalje se riječnik koji za P i T val ima polje interesnih blokova i vrhova koji ga određuju.



Slika 4.12 – pokretni prosjeci i interesni blokovi koji zadovoljavaju uvjete



Slika 4.13 – označeni vrhovi Elgendijevim algoritmom uz lutajuću nutlu liniju



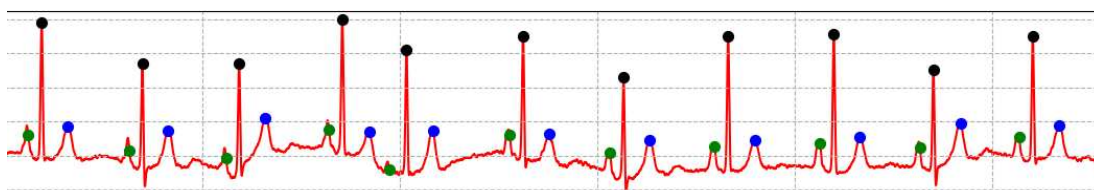
Slika 4.14 – označeni vrhovi korištenjem Elgendijevog algoritma, ista snimka kao i na slici 4.4

4.2.4. Morfološka analiza

Morfološka analiza izvodi se pozivom funkcije *analyze* iz modula *morphological_analysis*.

```
analyze(ecg, qrs, p_t_data, fs, plot=False)
```

Osim izvlačenja morfoloških značajki, u ovoj se funkciji poboljšava detekcija vrhova. Zbog nesavršenosti algoritama dobivene točke iz prethodnih koraka u programu se ne pogađaju nužno točno vrhovi. Zato se u ovoj funkciji samo postavljaju R, T i P vrhovi na poziciju maksimalne vrijednosti signala u širini svoga vala. Izvorna snimka signala na slici 4.15 ista je kao i na slikama 4.9 i 4.14. Promatrajući sve tri slike vidi se rezultat postupka ispravljanja pozicija vrhova na izvornoj snimci signala. Pozicije vrhova R, T i P valova na slici 4.15 točnije su postavljene nego na slikama 4.9 i 4.14. Nakon toga, dobivaju se morfološke značajke signala pozivanjem odgovarajućih funkcija. U tablici 4.16 prikazani su rezultati nakon provedenih dosadašnjih koraka programa. Dijagnoze su zapisane kao brojevi koji odgovaraju nabranjanju vrijednosti ciljne oznake u potpoglavlju *Podaci*.



Slika 4.15 – označeni vrhovi kao i na snimkama na slikama 4.7 i 4.4

diagnosis	age	sex	rate	r_amplitude	r_r_interval	t_amplitude	t_width	p_amplitude	p_width	merged_p_t	p_r_interval	s_t_segment	p_t_interval
7	74	1	100.0	48.0	0.6	94.83	0.07	56.67	0.06	0	0.02	0.29	0.44
1	49	2	78.0	905.44	0.77	158.79	0.09	63.51	0.08	0	0.16	0.13	0.4
2	81	2	114.0	864.43	0.53	31.9	0.08	36.41	0.08	7	0.16	0.12	0.4
2	45	1	105.46	933.5	0.57	37.89	0.08	82.9	0.07	5	0.17	0.1	0.39
6	53	1	120.0	604.91	0.5	197.0	0.09	91.18	0.06	1	0.12	0.05	0.29
7	29	1	64.29	849.7	0.88	297.55	0.09	42.05	0.06	0	0.1	0.1	0.32
2	74	1	120.0	734.97	0.51	105.54	0.07	129.53	0.06	1	0.11	0.12	0.36
8	80	2	67.21	933.43	0.89	39.65	0.07	5.78	0.08	0	0.15	0.15	0.42
2	81	1	60.0	798.11	1.01	192.12	0.09	75.56	0.06	0	0.19	0.17	0.48
7	82	1	72.0	690.05	0.85	29.22	0.06	40.2	0.07	0	0.13	0.04	0.29
4	87	1	66.67	863.01	0.89	297.74	0.1	28.57	0.06	0	0.18	0.06	0.4
6	74	1	71.46	453.67	0.86	208.83	0.09	91.99	0.08	0	0.13	0.13	0.38

Tablica 4.16 – izlučene morfološke značajke snimke elektrokardiograma za prvih 12 pacijenata

4.3. Vrednovanje i rezultati

Za klasificiranje su korišteni algoritmi opisani u trećem poglavlju: stroj s potpornim vektorima i logistička regresija. Korišteni su gotovi algoritmi iz modula *scikit-learn* [20]: *linear_model.LogisticRegression* za klasifikator koji koristi logističku regresiju i *svm.SVC* za klasifikator baziran na algoritmu stroja s potpornim vektorima.

Sljedeći korak bila je optimizacija parametara za klasifikatore. Za rješenje tog problema se sljedeće. Prvo je cijeli skup podataka podijeljen u skup za učenje i skup za testiranje. Dvadeset i pet posto skupa podataka nalazi se u skupu za testiranje, dok se ostatak nalazi u skupu za učenje. U isječku koda 4.17 vidi se postupak učitavanja podataka, razdjeljivanja na attribute i ciljnu oznaku te podjela skupa na skup za učenje i skup za testiranje.

```
data = pd.read_csv(export_file_main)

label = "diagnosis"

x = np.array(data.drop([label], 1))
y = np.array(data[label])

x_develop, x_test, y_develop, y_test = train_test_split(x, y,
test_size=0.25)
```

Isječak koda 4.17 – podjela na skup za učenje i skup za testiranje

Razred *GridSearchCV*, iz modula *scikit-learn* [20], implementira optimizaciju parametara algoritma za strojno učenje koristeći unakrsnu validaciju s k preklopa. Ovaj razred kao argumente u konstruktor prima: klasifikator za koji treba naći optimalne parametre, listu skupova parametara koje treba isprobati, mjera prema kojoj se vrednuje klasifikator i broj preklopa. Naredbom *fit* razreda *GridSearchCV* započinje optimizacija parametara. Naredba *fit* prima skup atributa (ulazne vrijednosti u klasifikator) i skup odgovarajućih ciljnih oznaka za ulaze. Unakrsnom validacijom se dobiva procjena mjere za vrednovanje klasifikatora u svakoj kombinaciji parametara. Unakrsna validacija s k preklopa, u ovom je slučaju k , jednak pet, provodi se na sljedeći način. Skup podataka, u ovom slučaju skup podataka za učenje, podijeli se na k jednako velikih skupova. U svakoj iteraciji stvara

se novi skup za učenje od četiri od k skupa i novi skup za validaciju koji sadrži preostali skup. Pomoću skupa za učenje razvija se model koji se pomoću skupa za validaciju vrednuje. Taj se postupak ponavlja onoliko puta koliko je preklopa tako da svaki od k skupova jednom bude korišten za vrednovanje. Konačno se model vrednuje gledajući rezultat vrednovanja u svakoj iteraciji. Zbog ovoga može se reći da početna podjela skupa podataka dijeli podatke na skup za razvoj i skup za testiranje, a skup za razvoj kasnije se dijeli na skup za učenje i skup za validaciju. Optimizacija parametara modela odvija se tako da se za svaku kombinaciju parametara iz ponuđenog skupa parametara napravi unakrsna validacija s k preklopa. Konačno, ona kombinacija parametara za koju je model najbolje vrednovan odabrana je kao optimalan izbor i s njom se stvara konačni klasifikator. U isječku koda 4.18 vidi se izvedba optimizacije parametara za stroj s potpornim vektorima, a na slici 4.19 ispis tog isječka koda. Parametar C označava vrijednost za meku granicu. Parametar *class_weight* ima vrijednost *balanced* što znači da algoritam koristi vrijednosti ciljne značajke kako bi prilagodio težine razreda obrnuto proporcijonalno učestalosti razreda u ulaznim podacima. *Kernel* označava koja se jezgrena funkcija koristi, u ovom slučaju radijalna bazna funkcija. Zadnji parametar, parametar *gamma*, određuje koeficijent za jezgrenu funkciju. Vrijednost parametra *gamma* je *scale* što znači da je vrijednost parametra *gamma* jednaka $1/(n_značajki * \text{varijanca ulaznih uzoraka})$.

```
svm_parameters = [{'kernel': ['rbf'],
                    'class_weight': ['balanced', None],
                    'C': [1, 2, 5, 8, 10, 12, 15, 20],
                    'gamma': ['scale', 'auto']},
                  {'kernel': ['linear'],
                    'class_weight': ['balanced', None],
                    'C': [1, 2, 5, 8, 10, 12, 15, 20]}]

svm_clf = GridSearchCV(svm.SVC(), svm_parameters, scoring='f1',
                       cv=5)
svm_clf.fit(x_develop, y_develop_binary
)
print("Best parameters for support vector machine")
print(svm_clf.best_params_)
```

Isječak koda 4.18 – optimizacija parametara za stroj s potpornim vektorima

```
Best parameters for support vector machine
{'C': 15, 'class_weight': 'balanced',
```

```
'gamma': 'scale', 'kernel': 'rbf'}
```

Ispis 4.19 – ispis rezultata pokretanja isječka koda 4.18

Naučeni model klasificira skup podataka za testiranje, nasumično odabranih dvadeset i pet posto izvornog skupa podataka. Te se klasifikacije spremaju u varijablu y_pred i koriste se za vrednovanje modela. U ovom završnom radu binarni klasifikatori su se vrednovali mjerom točnosti klasifikacije i mjerom F1. Točnost klasifikacije (*engl. accuracy*) udio je točno klasificiranih uzoraka u skupu svih uzoraka [10], izraz 4.20. Mjera F1 harmonijska je sredina preciznosti i odziva [10], izraz 4.23. Preciznost (*engl. precision*) je udio pozitivno klasificiranih uzoraka u skupu pozitivno klasificiranih uzoraka [10], izraz 4.21. Odziv (*engl. recall*) je udio pozitivno klasificiranih uzoraka u skupu svih pozitivnih uzoraka [10], izraz 4.22. Za višerazrednu klasifikaciju klasifikatori su se vrednovali makro-prosjekom točnosti i F1-mjere. Makro-prosjek znači da se mjera izračunava za svaki razred u višerazrednoj klasifikaciji pa se radi uprosječivanje kroz razrede [10]. Na isječku iz koda 4.24 vidi se implementacija vrednovanja stroja s potpunim vektorima s optimiziranim parametrima, a na 4.25 ispis rezultata pokretanja tog isječka.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.20)$$

$$P = \frac{TP}{TP + FP} \quad (4.21)$$

$$R = \frac{TP}{TP + FN} \quad (4.22)$$

$$F1 = \frac{2PR}{P + R} \quad (4.23)$$

```
y_pred = svm_clf.predict(x_test)

svm_clf_acc = metrics.accuracy_score(y_test_binary, y_pred)
print('Accuracy = {}'.format(svm_clf_acc))
svm_clf_f1 = metrics.f1_score(y_test_binary, y_pred)
print("F1 score = {}".format(svm_clf_f1))
```

Isječak koda 4.24 – vrednovanje mjerama točnosti i mjere F1

Accuracy = 0.7416719044626021
F1 score = 0.4804045512010114

Ispis 4.25 – ispis rezultata pokretanja isječka koda 4.24

Isprobana su dva klasifikacijska problema: višerazredna klasifikacija srčanog ritma od mogućih devet i binarna klasifikacija normalnog sinus ritma ili aritmije. Tablica 4.26 prikazuje rezultate vrednovanja za binarnu klasifikaciju, a tablica 4.27 prikazuje rezultate vrednovanja za višerazrednu klasifikaciju.

Binarna klasifikacija	Točnost klasifikacije	F1-mjera
Logistička regresija	0.868007	0.403409
Stroj s potpornim vektorima	0.741719	0.480405

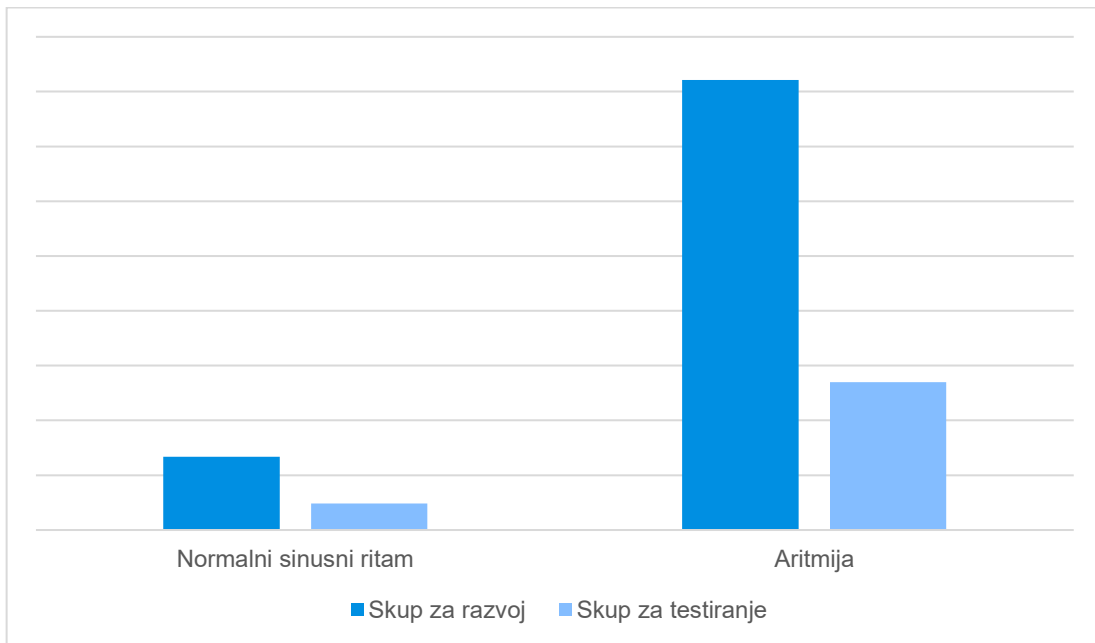
Tablica 4.26 – vrednovanje klasifikatora za binarnu klasifikaciju

Višerazredna klasifikacija	Točnost klasifikacije	F1-mjera
Logistička regresija	0.4770585	0.3308029
Stroj s potpornim vektorima	0.3412948	0.2971461

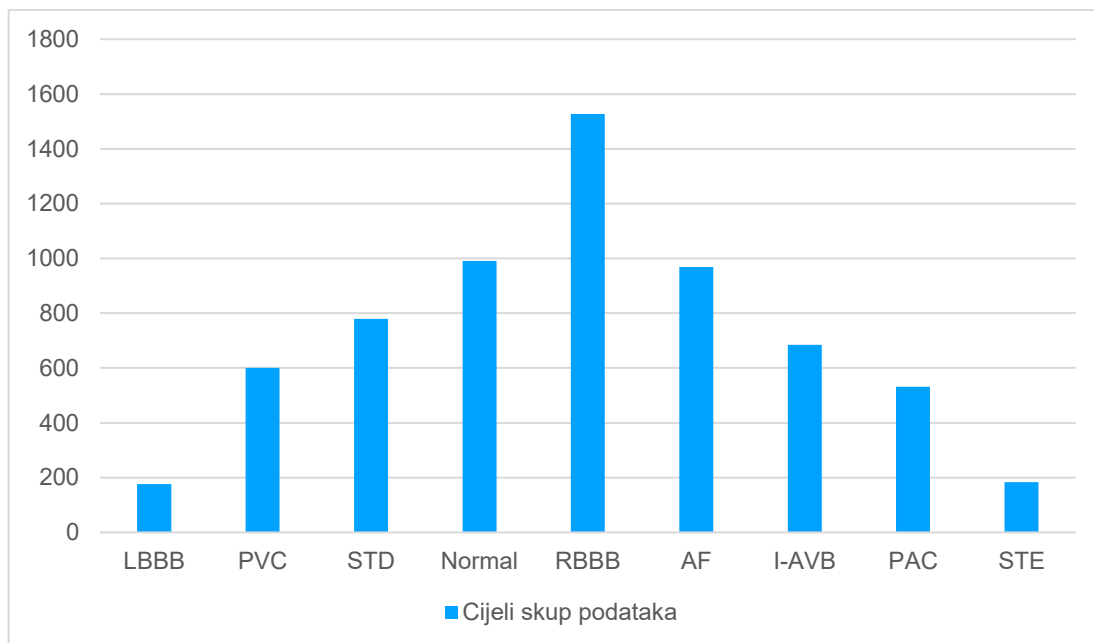
Tablica 4.27 – vrednovanje klasifikatora za višerazrednu klasifikaciju

Iz tablice 4.26 očita je razlika između mjere točnosti klasifikacije i F1-mjere. To je zbog neravnomjerno raspodijeljenih razreda u skupu podataka za binarnu klasifikaciju. Na slici 4.28 prikazan je stupičasti graf koji opisuje razdiobu uzoraka s dijagnozom normalnog sinusnog ritma i uzoraka s dijagnozom aritmije u skupu za razvoj i skupu za testiranje. Očito je kako aritmije prevladavaju u oba skupa. Zbog

toga mjeru točnosti klasifikacije možemo zanemariti jer i nevaljani klasifikator koji uvijek klasificira uzorak kao aritmiju može postići dobru točnost. Gledajući mjeru F1 puno bolje se vidi stvarno vrednovanje binarnog klasifikatora. Kod višerazredne klasifikacije to očito nije toliko problem zbog pogodnije razdiobe. Na slici 4.29 prikazan je stupičasti graf koji opisuje razdiobu uzoraka prema dijagnozi u cijelom skupu podataka. Iako se ne radi o jednolikoj razdiobi, već normalnoj, prilagođenija je klasifikaciji nego razdioba uzoraka u binarnoj klasifikaciji. Zbog toga je kod višerazredne klasifikacije razlika između mjere točnosti i F1-mjere manja nego kod binarne klasifikacije. Uspoređujući dobivene rezultate s rezultatima na rang listi, kojoj se može pristupiti na stranici izazova [13], rezultati ovog završnog rada pripadaju donjoj polovici.



Slika 4.28 – razdioba uzoraka u binarnoj klasifikaciji



Slika 4.29 – razdioba uzoraka cijelog skupa podataka u višerazrednoj klasifikaciji

5. Zaključak

Rezultati ovog završnog rada razumljivi su. Naravno da će binarna klasifikacija dati bolju točnost, no zašto točnost klasifikacije devet razreda nije dala tako veliku točnost nije odmah uočljivo. Sve se svodi na korištene algoritme i njihovu implementaciju. Algoritmi za detekciju ključnih točaka na snimci mogu se nadograditi i poboljšati. Počevši, recimo, od pomnijeg proučavanja parametara koji se koriste. Kod algoritma za detekciju QRS kompleksa može se obratiti veća pozornost pri izboru širine prozora kod integracije pomičnim prozorom, udaljenosti vrhova kod traženja vrhova prije odlučivanja o njima. Također, može se nadograditi implementacija istog algoritma korištenjem ostalih značajki izvorne implementacije. Kod algoritma za detekciju P i T valova puno je parametara koje se može još finije odabrati. Širina prihvatljivog bloka, minimalna i maksimalna udaljenost prihvatljivog bloka od vrha R vala. Mogu se dodati dodatne morfološke značajke koje bi pomogle pri klasifikaciji. Značajka koja opisuje regularnost srčanog ritma mogla bi poboljšati rezultate. Umjesto izlučivanja QT intervala mogao bi se izlučiti ispravljeni QT interval (QTc). Također, proučavanje odabira značajki za određene klasifikatore valjan je smjer nadogradnje. Naposljetku, bolje rezultate moglo bi se postići korištenjem naprednijih modela i metoda kao što su neuronske mreže i duboko učenje. Daljni korak u primjeni može biti u nosivoj tehnologiji, npr. pametni satovi.

Zadatak klasifikacije srčanih ritmova iz snimaka 12-kanalnog elektrokardiograma pokazao se zahtjevnim, ali vrlo zanimljivim. Primjena stečenih i novih znanja iz računarstva na područje koje nije tako usko vezano uz računarstvo jedno je od vrjednijih dijelova ovog završnog rada.

6. Literatura

- [1] World Health Organization, "Cardiovascular diseases". Dostupno na: <https://www.who.int> [Pristupljeno 30. svibnja 2020.]
- [2] T. B. Garcia i N. E. Holtz, "12-Lead ECG The Art of Interpretation", Jones and Bartlett Publisher, 2001
- [3] Wikipedia, "Desno srce". Dostupno na: <https://bs.wikipedia.org> [Pristupljeno 31. svibnja 2020.]
- [4] Life in the fast lane, "ECG Lead positioning". Dostupno na: <https://litfl.com> [Pristupljeno 2. lipnja 2020.]
- [5] Cables and sensors, "12-Lead ECG Placement Guide with Illustrations". Dostupno na: <https://www.cablesandsensors.com> [Pristupljeno 2. lipnja 2020.]
- [6] J. Pan i W. J. Tompkins, "A Real-Time QRS Detection Algorithm", 1985.
- [7] M. Elgendi, B. Eskofijer i D. Abbot, "Fast T Wave Detection Calibrated by Clinical Knowledge with Annotation of P and T Waves", 2015.
- [8] I. H. Witten, E. Frank, M. A. Hall, „Data Mining Practical Machine Learning Tools and Techniques“, Morgan Kaufmann Publisher is an imprint od Elsevier, 2011
- [9] E. Alpaydin, „Introduction to Machine Learning third edition“, The MIT Press, 2014
- [10] J. Šnajder, „Strojno učenje – natuknice za predavanja“, UNIZG FER, 2017.
- [11] J. Šnajder, „Strojno učenje – videopredavanja“, UNIZG FER, 2018.
- [12] F. F. Liu, C. Y. Liu, L. N. Zhao, X. Y. Zhang, X. L. Wu, X. Y. Xu, Y. L. Liu, C. Y. Ma, S. S. Wei, Z. Q. He, J. Q. Li and N. Y. Kwee. An open access database for evaluating the algorithms of ECG rhythm and morphology abnormal detection. Journal of Medical Imaging and Health Informatics, 2018, 8(7): 1368–1373.
- [13] PhysioNet/CinC Challenges, "Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020". Dostupno na: physionetchallenges.github.io [Pristupljeno 10. lipnja 2020.]

- [14] ICBEB 2018, “The 7th International Conference on Biomedical Engineering and Biotechnology”. Dostupno na: 2018.icbeb.org [Pristupljeno 7. lipnja 2020.]
- [15] Python, “The official home of the Python Programming Language”. Dostupno na: <https://www.python.org> [Pristupljeno 2. lipnja 2020.]
- [16] NumPy, “The fundamental package for scientific computing with Python”. Dostupno na: <https://numpy.org> [Pristupljeno 5. lipnja 2020.]
- [17] SciPy, “Python-based ecosystem of open-source software for mathematics, science, and engineering”. Dostupno na: <https://scipy.org> [Pristupljeno 5. lipnja 2020.]
- [18] Pandas, “Python Data Analysis Library”. Dostupno na: <https://pandas.pydata.org>
- [19] Matplotlib, “Python plotting. Dostupno na: <https://matplotlib.org> [Pristupljeno 5. lipnja 2020.]
- [20] Scikit-learn, “Machine learning in python”. Dostupno na: <https://scikit-learn.org> [Pristupljeno 5. lipnja 2020.]

Sažetak

Kardiovaskularne bolesti najrašireniji su i najčešći uzrok smrti na svijetu. Razvoj sustava za klasifikaciju srčanih ritmova iz snimaka rada srca ima široku mogućnost primjene. U ovom završnom radu istražuje se područje kardiologije, analize signala te strojnog učenja. Domensko znanje iz kardiologije ključno je za uspješno pronalaženje i izvlačenje značajki iz snimke EKG-a. Istražuju se algoritmi za analizu snimke signala EKG-a: Pan-Tompkinsov algoritam za pronalazak QRS kompleksa i Elgendijev algoritam za pronalazak T i P valova. Za klasifikaciju su korišteni algoritmi logističke regresije i stroja s potpornim vektorima. Prikazani su rezultati za dva klasifikacijska problema, binarna klasifikacija normalnog sinusnog ritma i aritmije te višerazredna klasifikaciju devet srčanih ritmova. Iz rezultata se vidi prostor za napredak rješenja, bilo to u nadogradnji algoritama za analizu signala ili korištenju drugih algoritama strojnog učenja.

Abstract

Cardiovascular diseases are the leading cause of death globally. The development of a system for classification of heart rhythms from 12-lead ECG-s has a wide range of use. This bachelor thesis researches the field of cardiology, signal analysis and machine learning. Knowledge of cardiology is vital for successfully demarcating and extracting features from electrocardiograms. Signal analysis algorithms used are the following: Pan-Tompkins algorithm for QRS complex detection and Elgendi's algorithm for T and P wave detection. Logistic regression and support vector machine algorithms are used for classification. Achieved results on two classification problems, binary classification of normal sinus rhythm and arrhythmia and multiclass classification of nine different heart rhythms, are presented in this thesis. Results indicate some space for improvement of the solution which can be explored by improving the implementation of signal analysis algorithms or by using other machine learning algorithms.