

**SVEUČILIŠTE U SPLITU
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I
BRODOGRADNJE**

DIPLOMSKI RAD

**PRIMJENA TEHNIKA STROJNOG UČENJA NA
PODATKE IZ DOMENE MEDICINE**

Bruno Grbavac

Split, rujan 2023.



Preddiplomski studij: **Naziv studija**

Smjer/Usmjerenje: **Naziv smjera/usmjerenja**

Oznaka programa: **110**

Akademska godina: **2016./2017.**

Ime i prezime: **IME PREZIME**

Broj indeksa: **xxxxxxxxxxxxxx**

ZADATAK ZAVRŠNOG RADA

Naslov: **NASLOV RADA**

Zadatak: Prilikom prijave završnog rada mentor definira temu i radni naslov, a točan naslov rada utvrđuje se po završetku rada. Završni se rad predaje mentoru u elektroničkoj formi. Rad potpisuje mentor. Izradu završnog rada ocjenjuje mentor. **Ovu stranicu piše mentor prije predaje rada nakon što je utvrđen konačni naslov rada. Zatamnjeni dijelovi su promjenjivi, a prilikom ispisa zatamnjene izbrisati.**

Prijava rada: **02.10.2016. (početak semestra u kojem se prijavljuje rad)**

Rok za predaju rada: **14.02.2016. (deset dana prije završetka semestra u kojem je rad prijavljen)**

Rad predan: **Datum**

Datum obrane: **Datum**

Mentor:

izv. prof. dr. sc. Ime Prezime,

IZJAVA

Ovom izjavom potvrđujem da sam diplomski rad s naslovom **Primjena tehnika strojnog učenja na podatke iz domene medicine** pod mentorstvom **izv. prof. dr. sc. Tonija Perkovića** pisao samostalno, primjenivši znanja i vještine stečene tijekom studiranja na Fakultetu elektrotehnike, strojarstva i brodogradnje, kao i metodologiju znanstveno-istraživačkog rada, te uz korištenje literature koja je navedena u radu. Spoznaje, stavove, zaključke, teorije i zakonitosti drugih autora koje sam izravno ili parafrazirajući naveo u diplomskom radu citirao sam i povezao s korištenim bibliografskim jedinicama.

Student

Bruno Grbavac

Sadržaj

1. UVOD	1
2. STROJNO UČENJE	2
2.1. Problemi strojnog učenja	4
2.1.1. Klasifikacija	4
2.1.2. Regresija	4
2.1.3. Detekcija anomalija	5
2.1.4. Grupiranje (engl. <i>clustering</i>)	6
2.1.5. Poduprto učenje (engl. <i>reinforcement learning</i>)	6
2.2. Model	7
2.3. Podjela modela strojnog učenja	8
2.3.1. Nadzirano učenje (engl. <i>supervised learning</i>)	8
2.3.2. Nenadzirano učenje (engl. <i>unsupervised learning</i>)	8
2.3.3. Polu-nadzirano učenje	8
2.3.4. Poduprto učenje	8
2.3.5. Evolucijsko ili genetsko učenje	8
2.3.6. Umjetne neuronske mreže (engl. <i>ANN - artificial neural networks</i>)	9
2.3.7. Učenje s više modela (engl. <i>ensamble learning</i>)	10
2.3.8. Učenje temeljeno na pojedinim ulazima (engl. <i>instance based learning</i>)	12
2.3.9. Algoritmi redukcije dimenzija	12
2.3.10. Hibridni algoritmi	12
2.4. Strojno učenje u medicini	12
3. ISTRAŽIVANJE	17
3.1. Apnea spavanja	17
3.1.1. Klasifikacija ozbiljnosti oboljenja apneje	17
3.1.2. Opstruktivna apnea	18
3.1.3. Centralna apnea	20
3.2. Polisomnografija	21
4. PODACI	23
4.1. Prikupljanje podataka	23

4.1.1. Philips Alice PDX	23
4.2. Obrada i interpretacija	25
4.2.1. Formati pohrane	28
5. OKRUŽENJE	31
5.1. Anaconda	31
5.2. Google Colab	31
5.3. Azure ML	32
5.4. Jupyter	32
6. IMPLEMENTACIJA	33
6.1. Automatizacija i optimizacija	33
6.1.1. Papermill	35
6.2. Random forest	37
6.3. Konvolucijske neuronske mreže	40
6.4. LSTM (engl. <i>Long Short-Term Memory</i>) mreže	42
6.5. Klasifikacija faze sna	46
6.5.1. Analiza značajki	46
6.6. Predviđanje AHI indeksa	53
6.7. Predviđanje razine SpO_2	55
7. ZAKLJUČAK	57
LITERATURA	58
POPIS OZNAKA I KRATICA	60
SAŽETAK	61
SUMMARY	62

1. UVOD

Od svojih samih začetaka sredinom prošloga stoljeća umjetna inteligencija i strojno učenje teže se što rapidnije primijeniti na poboljšanje ljudskih života, a samim time neizbjegna je konvergencija područja iz navedene sfere računalne znanosti i medicine.

Iako je sama klinička primjena ovakvih sustava skoro pola stoljeća bila ograničena i usporena ograničenjima samog hardvera, danas - kao i čitavo prethodno desetljeće, možemo pouzdano reći kako svjedočimo dugo iščekivanom došašću sustava temeljenih na umjetnoj inteligenciji i strojnom učenju u svim sferama života pa tako i u, usudio bi se reći najplemenitijem području ljudskog djelovanja - medicini. Neizostavan preduvjet primjene umjetne inteligencije je podatkovna znanost koja kao područje računalne znanosti i statistike također bilježi znakoviti rast popularnosti kako u samim profesionalnim tako i u akademskim krugovima. Kroz ovaj rad pokušati ću prikazati primjenu tehnika iz sfere strojnog učenja i podatkovne znanosti na podacima prikupljenim i korištenim u sklopu projekta kojeg provodi Znanstveno-istraživačka baza splitskog Centra za medicinu spavanja pri Kliničko Bolničkom Centru Split sa ciljem detekcije apneje i hipopneje i istočnih komorbiditeta tih poremećaja u disanju tokom sna.

2. STROJNO UČENJE

Učenje kod ljudi možemo definirati kao proces sticanja novih ili izmjene postojećih znanja i vještina. Strojno učenje je u svojoj srži grana **umjetne inteligencije** kao područja računarne znanosti, čiji sustavi mogu na osnovu priloženih podataka ostvariti učenje. Sam pojam se prvi put pojavljuje 1959. godine kada ga Arthur Samuel definira kao područje čije spoznaje omogućuju računalima uspješno izvršavanje zadaća bez da je proces rješavanja prethodno programirao čovjek. [2]

*"Za računalni program kažemo da uči ako se, kroz iskustvo **I** vezano uz zadatak **Z**, ocjene performansi **P** izvršavanja zadatka **T** poboljšavaju rastom **I**."* - Tom Mitchell [2]

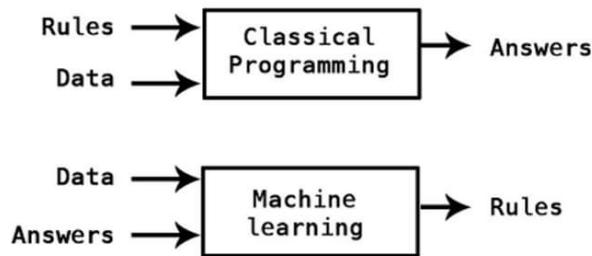


Slika 2.1: prof. Arthur Samuel i prof.dr.sc. Tom Mitchell (slike preuzete sa Carnegie Mellon University i Standford University¹)

"Strojno učenje proces je u kojem računalni sustavi uče iz podataka koristeći razne algoritme kako bi izvršavali zadatke bez da je takvo znanje u njih eksplicitno programirano." [1]

Razlika između ovih sustava i klasičnih sustava za izvještaje (engl. "dashboard") koji iz postojećih podataka računaju i prikazuju određene karakteristike je upravo što su **modeli** strojnog učenja entiteti s mogućnosti predviđanja evolucije podataka promjenom zavisnih varijabli. [3].

¹Carnegie Mellon University: Tom Mitchell, s Interneta: <http://www.cs.cmu.edu/~tom/>, 11.09.2023., Standford Engineering Forum: Arthur Samuel, s Interneta: <https://forum.stanford.edu/people/arthur-samuel>, 11.09.2023.



Slika 2.2: Razlika strojnog učenja i klasičnih programa. (slika preuzeta sa GitBook²).

"Model je bilo koja funkcija s moći predviđanja." [3]

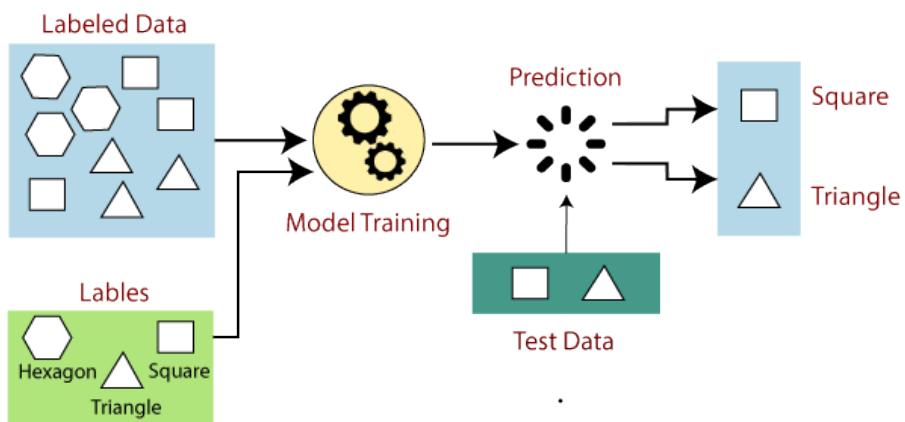
Modeli strojnog učenja imaju kao glavnu zadaću pronaći uzorke i poveznice u podacima na osnovu kojih odlučuju odnosno izvršavaju zadaću. Učenje se odvija kroz modifikaciju vlastitih postupaka na način da se teži ka postizanju veće **preciznosti** kao brojčane vrijednosti izračunate na osnovu udjela točnih odluka u ukupnom broju odluka. [4][2]

²A Few Useful Things to Know About Machine Learning, s Interneta: <https://xzhu0027.gitbook.io/blog/machine-learning/untitled/ml>, 11.09.2023.

2.1. Problemi strojnog učenja

2.1.1. Klasifikacija

Problem kod kojega je kao izlaz potrebno ulaz svrstati u jednu od konačnog broja definiranih klasa. Ovisno o broju klasa radi se o **binarnoj klasifikaciji** (npr. 'Postoje li osobe na slici?') Odgovor je binaran 'Da' ili 'Ne'.) ili **višeklasnoj** (engl. *multiclass*) klasifikaciji (npr. 'Koja vrsta voća je na slici?' Odgovor može biti 'jabuka', 'kruška', 'banana' ili 'nema voća na slici'.) [2]

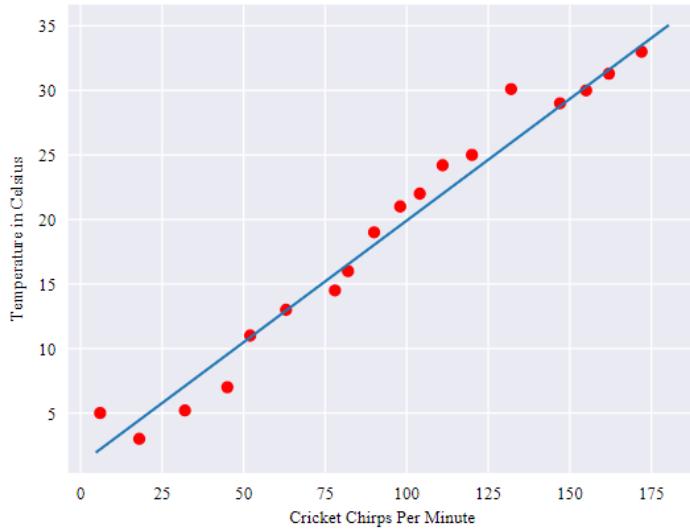


Slika 2.3: Skica klasifikacije geometrijskih oblika. (slika preuzeta sa K21Academy³)

2.1.2. Regresija

Algoritmi koji se bave regresijom bave se prvenstveno kontinuiranim numeričkim ulazima sa ciljem predviđanja odgovora na pitanja 'koliko'. (npr. predviđanje promjene temperature učionice). [2]

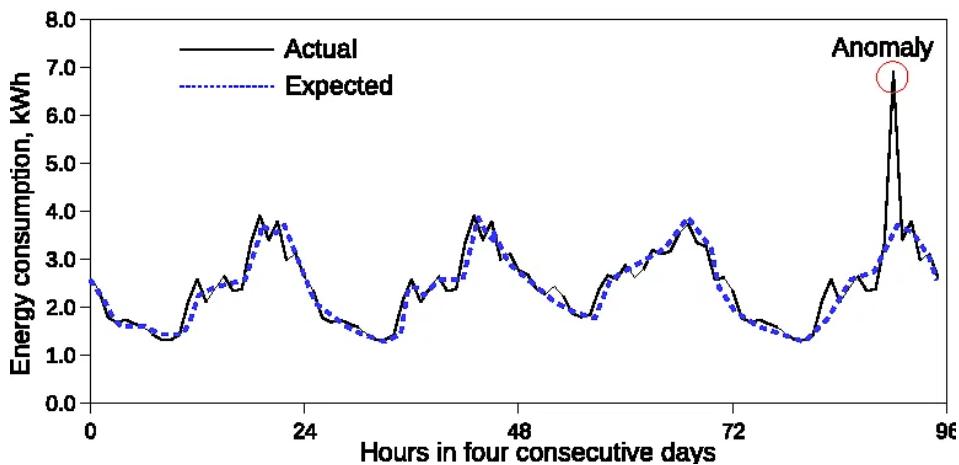
³K21Academy: Amazon SageMaker Built-in Algorithms – Introduction, s Interneta: <https://k21academy.com/amazon-web-services/aws-ml/amazon-sagemaker-algorithms/>, 11.09.2023.



Slika 2.4: Linearna regresija učestalosti glasanja cvrčka ovisno o temperaturi (slika preuzeta sa Google Developers⁴).

2.1.3. Detekcija anomalija

Problem kod kojeg je potrebno da model analizira i uoči uzorke unutar signala te reagira/istakne diskontinuitet u istima - detekcija neuklapajućih signala (engl. *outliers*). (npr. Banka treba detektirati anomaliju u transakcijama korisnika kako bi se spriječila krađa kartice.)



Slika 2.5: Detekcija anomalije visoke potrošnje električne struje (slika preuzeta sa ProjectPro⁵).

⁴Google: Descending into ML: Linear Regression, s Interneta: <https://developers.google.com/machine-learning/crash-course/descending-into-ml/linear-regression>, 11.09.2023.

2.1.4. Grupiranje (engl. *clustering*)

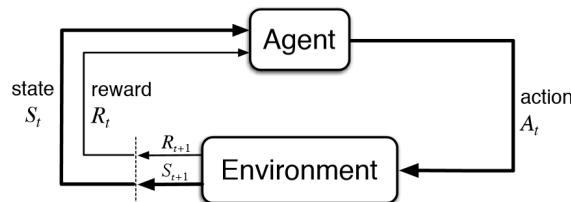
Problemi ovog tipa naizgled su slični klasifikaciji no razlika je što ovi algoritmi kao zadaću imaju analizom strukture podataka uočiti skrivene obrasce i definirati same kategorije u koje zatim svrstavaju ulaze, dok su klasifikacije one prethodno zadate.



Slika 2.6: Grupiranje potrošača ovisno o važnosti i prihodima (slika preuzeta sa TowardsAI⁶).

2.1.5. Poduprto učenje (engl. *reinforcement learning*)

Ovakvi sustavi uče rješavanje problema po principu pokušaja i pogreški sa postavljenim sustavom nagrada i kazni bez da je prethodno definiran cilj zadatka. Ovakva rješenja popularna su u igrama i problemima poput regulacije temperature ili vlage zraka. [2]



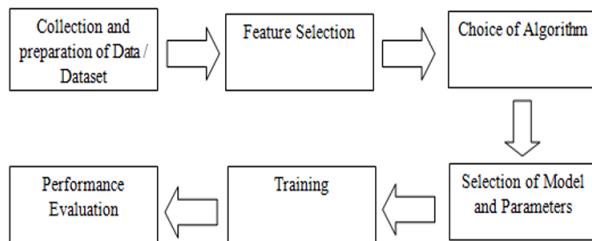
Slika 2.7: Prikaz modela poduprtog učenja kao automata (slika preuzeta sa Encord⁷).

⁵ProjectPro: How to do Anomaly Detection using Machine Learning in Python?, s Interneta: <https://www.projectpro.io/article/anomaly-detection-using-machine-learning-in-python-with-example/555>, 11.09.2023.

⁶TowardsAI: Fully Explained Hierarchical Clustering with Python, s Interneta: <https://pub.towardsai.net/fully-explained-hierarchical-clustering-with-python-ebb256317b50>, 11.09.2023.

2.2. Model

Općeniti model strojnog učenja da se prikazati kroz šest koraka koji su u određenom obliku prisutni kod velike većine ili svih modela. [2]



Slika 2.8: Općeniti model strojnog učenja. [2]

1. **Prikupljanje i obrada podataka** za prvi zadatak ima sakupiti iz različitih izvora (npr. web, vlastiti senzori, kupnja podatkovnih skupova). Zatim je potrebno prikupljene podatke obraditi kako bi bili oblika kojeg zahtjeva korišteni algoritam (npr. otklanjanje šuma, skaliranje vrijednosti varijable, promjena frekvencije signala).
2. **Odabir značajki** odnosi se na odabir samo onih značajki/varijabli/stupaca koje su nam potrebne odnosno relevantne za daljnji rad, dok je ostale potrebno ukloniti.
3. **Odabir algoritma** koji će za postavljeni problem pružiti najbolje rezultate je imperativ s obzirom da nisu svi algoritmi prikladni za rješavanje svih problema.
4. **Postavljanje modela odabirom parametara** odnosi se na postavljanje programa za automatsko parametara odabranoga algoritma ili ručno postavljanje istih na vrijednosti koje su procijenjene kao optimalne za postojeći problem.
5. **Treniranje** modela je korištenje dijela prikupljenih podataka kako bi se postavili razni interni parametri pod-funkcija odabranoga algoritma.
6. **Ocjena performansi** radi se prije same primjene modela, kako bi se utvrdila njegova uspješnost na ostatku prikupljenih podataka koje model nije vidio tokom faze treniranja. Performanse se procjenjuju računajući razne metrike poput **točnosti** (engl.*accuracy*) i **odziva** (engl. *recall*).

⁷Encord: Guide to Reinforcement Learning from Human Feedback (RLHF) for Computer Vision, s Interneta: <https://encord.com/blog/guide-to-rlhf/>, 11.09.2023.

2.3. Podjela modela strojnog učenja

Iako je desetljećima ustaljena podjela strojnog učenja na paradigme nadziranog i nenadziranog učenja, danas se ta podjela profilirala u deset kategorija. [2]

2.3.1. Nadzirano učenje (engl. *supervised learning*)

Kod ovog oblika učenja kroz skup podataka za trening dani su ispravni izlazi samog modela, takve podatke nazivamo **označenima** (engl. *labeled*). Stoga ovakvi modeli uče kroz iterativnu procjenu svoje podudarnosti s podacima iz skupa za treniranje (npr. učenje raspoznavanja različitih pasmina - date su slike svake pasmine). Ovakvo učenje nazivamo još učenjem poduprtim primjerima. Modeli nadziranog učenja dalje se dijele na klasifikaciju i regresiju pri čemu su izlazi klasifikacije diskretni, a regresije kontinuirani.

2.3.2. Nenadzirano učenje (engl. *unsupervised learning*)

Nenadziranim učenjem smatraju se oni oblici strojnog učenja koji ne koriste označene podatke za treniranje modela, tj. zadaća je da se tokom treniranja modela pronađu, kroz analizu strukture podataka, dotad nevidljivi postojeći uzorci iz kojih se uspostavljaju pravila za donešenje odluka. Ovakvi sustavi imaju primjenu u slučajevima kada nisu prethodno poznate kategorije izlaza.

2.3.3. Polu-nadzirano učenje

Ovaj oblik učenja pojavljuje se kod skupova podataka koji su djelomično "*labelirani*" zbog prirode podataka ili samog troška označavanja (označavanje je potrebno raditi ručno). Takvo učenje pruža benefite i podržanog i nepodržanog učenja te se koristi na problemima klasifikacije, predviđanja i regresije kao što su **generativni modeli**, **samotrenirajući modeli** te **prijenosni SVM** (engl. *support vector machine*).

2.3.4. Poduprto učenje

Kao što je već spomenuto u poglavlju 2.1.5. ovakav oblik učenja ostvaren je tako da se izlaz modela ocjenjuje ili kao ispravan ili kao neispravan te algoritam mora takvim grananjem doći do točnog izlaza te uspostaviti strukturu koja što većem udjelu ulaza pridružuje ispravan izlaz bez drugog usmjeravanja.

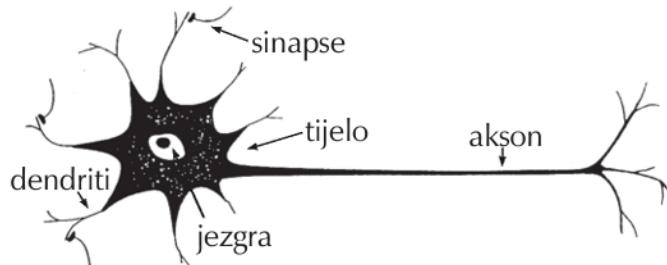
2.3.5. Evolucijsko ili genetsko učenje

Ovi algoritmi temelje se na ideji darvinizma i evolucije tj. **opstanka najjačih** (engl. *survival of the fittest*). Početna točka je **populacija** mogućih rješenja (modela) od kojih zatim samo dio, na osnovu veće uspješnosti rješavanja zadatka, stvara novu populaciju odnosno novi

skup mogućih rješenja nastalih funkcijom križanja (računa parametre novog modela na osnovu parametara modela roditelja) iz prethodne generacije.

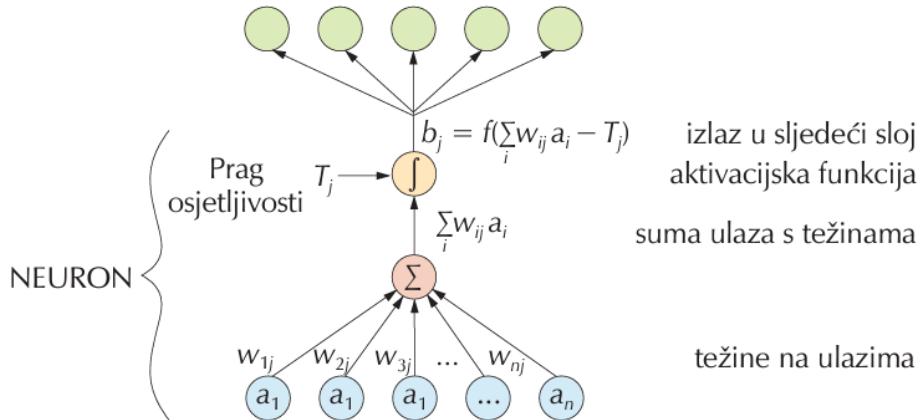
2.3.6. Umjetne neuronske mreže (engl. *ANN - artificial neural networks*)

Ovaj skup algoritama strojnog učenja temeljen je na stvarnim živčanim stanicama i njihovoj umreženošći te sposobnosti takve mreže da uči i zaključuje. Sami ljudski neuron sastoji se od **tijela stanice (soma)** koja je zapravo 'procesor' neurona, te se na osnovu obrade u tijelu donosi odluka o odašiljanju električnih podražaja prema susjednim neuronima i obliku tog podražaja. Tijelo također sadrži **jezgru (nukleus)** u kojoj su pohranjene informacije. Samo tijelo okruženo je **dendritima** - kratkim živčanim vlaknima preko kojih u somu dolaze impulsi s drugih neurona, dok su duge niti - **aksoni**, zaduženi za odašiljanje impulsa prema ostatku živčane mreže. Same stanice mreže povezane su **sinapsama** -stanicama koje izlučuju neurotransmitere i tako omogućuju prijenos impulsa sa aksona na dendrite (tokom prijenosa se signal mijenja - *sinaptička operacija*), dendriti signal puštaju prema somi gdje se on obrađuje - računa se izlazni signal i ažurira informacija u jezgri (*somatska operacija*) te se kao takav izlazni signal prosljeđuje kroz aksone prema ostatku mreže. Pojam umjetnih mreža u svom radu koji



Slika 2.9: Građa živčane stanice. [6]

opisuje računanje koristeći model umjetnog neurona **perceptron** 1943. uvode **McCulloh i Pitts**. Perceptroni umjetnih neuronskih mreža temeljeni su na sinaptičkim i somatskim funkcijama. Sastoje se od ulaznog sloja - ulaznih signala, koji se u jezgru tj. skriveni sloj uvoze kroz sinaptičku operaciju gdje se pojedini ulaz a_i **otežava (množi)** odgovarajućom **težinom** w_i .



Slika 2.10: Građa umjetnog neurona - perceptrona. [6]

Suma otežanih ulaza zbrojena sa vrijednošću koju nazivamo engleski **bias** se zatim uspoređuje sa **pragom aktivacije** neurona T_j . Ako je suma veća od praga **aktivacijska funkcija (nelinearna)** na osnovu nje računa izlaz koji se prosljeđuje ka sljedećem sloju.

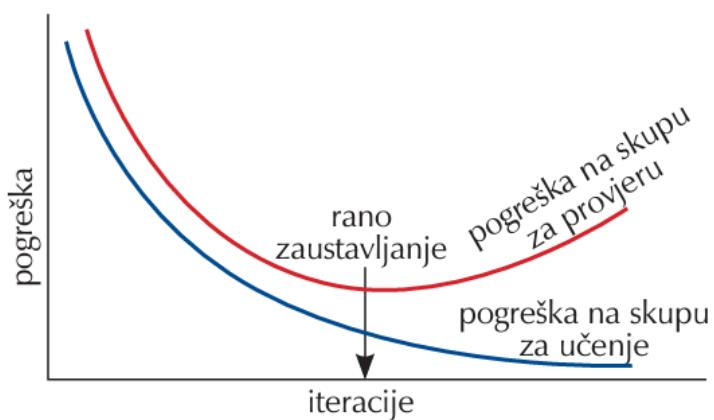
Sam proces učenja neuronskih mreža svodi se na **podešavanje težinskih parametara** w_i kako bi se smanjila **izlazna greška** - razlika između točne vrijednosti i one izračunate neuronskom mrežom. Podešavanje se vrši nekim od mnogih pravila učenja od kojih je najčešće korištena **unatražna propagacija izlazne pogreške**. Pravila se primjenjuju tako da se nakon svakog prolaska informacije kroz mrežu - izračuna, dobivena greška pokušava smanjiti. Kroz epohe treniranja - podešavanja, greška na skupu podataka za učenje težit će ka nuli (ukoliko imamo smislena pravila učenja) što ne mora biti dobro - jer tada model gubi **moć generalizacije** tj. ispravnog zaključivanja na primjerima ulaza koje do sada nije susreo kroz treniranje. Podešavanje parametara da **previše odgovaraju** skupu za treniranje nazivamo **pretreniranost (engl. overfitting)** stoga je potrebno odabrati nevelik broj epoha i testirati na skupu podataka odvojenih za test koji mu nisu predviđeni u treniranju.

2.3.7. Učenje s više modela (engl. *ensamble learning*)

Ovakvi modeli sastoje se od više individualnih modela koji rješavaju zajednički problem, postavljenih sekvencijalno (serijski) ili paralelno. Za razliku od običnih modela, u ovom se slučaju razvija više međusobno zavisnih ili nezavisnih (serijski ili paralelno) hipoteza pripadajućih modela čijim se sjednjnjem pokušava smanjiti otklon (engl. *bias*), varijanca odgovora (*bagging* - koristi se paralelni skup modela treniranih na nejednakim skupovima te se njihov prosjek uzima kao konačno predviđanje) ili samo poboljšanje predviđanja.

Biološki neuron		Umjetni neuron
	Prima ulazni signal putem dendrida (sinaptičke veze)	Prima ulaze (i) koji su određeni težinskim koeficijentima (w)
	Obrada signala u somi	Obrada ulaza, unutarnji prag – bias (b)
	Pretvara obrađeni ulaz u izlaz putem aksona	Pretvara ulaze u izlaz (prijenosna funkcija)
	Šalje informacije putem sinapsi do svih neurona s kojima je neuron povezan	Šalje informaciju prema izlazu i sljedećim neuronima

Slika 2.11: Usporedba perceptron i neurona. [6]



Slika 2.12: Općeniti prikaz izlazne pogreške u ovisnosti o broju epoha treniranja. [6]

2.3.8. Učenje temeljeno na pojedinim ulazima (engl. *instance based learning*)

Za razliku od drugih modela, ovakvi modeli ne daju nakon treniranja eksplisitnu funkciju izlaza, već pohranjuju svoju odluku o skupu podataka za treniranje, a samo se generaliziranje odgadja za kada se modelu predoče sami podaci za klasifikaciju (nazivaju se još i **lijeni** modeli). U trenutku kada se modelu predoči primjer kojeg dosad nije vidoio, on se funkcijom sličnosti usporedi s onima iz skupa za treniranje te se na osnovu toga donosi odluka. Ovakvi modeli popularni su kod problema grupiranja (npr. KMeans).

2.3.9. Algoritmi redukcije dimenzija

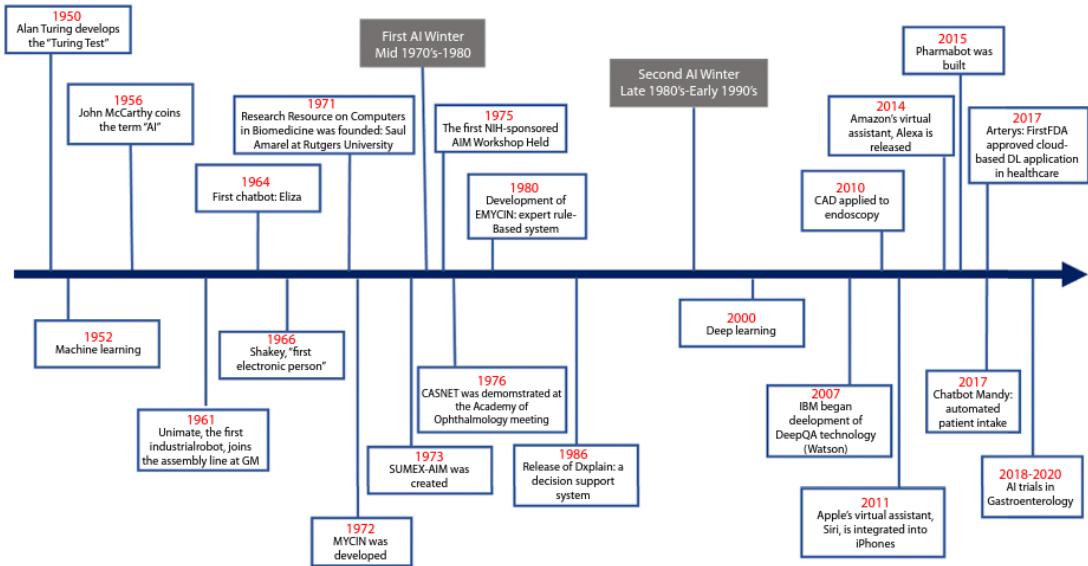
Kod velikih skupova podataka jedan od najvećih problema je efikasnost pri učenju, jer je taj proces jako usporen (zahtjeva osim vremena i velike količine drugih resursa) velikim brojem dimenzija ulaznog skupa tj. brojem različitih značajki koje su manje ili više bitne za konačan rezultat. Pod redukciju dimenzija spadaju svi algoritmi koji smanjuju broj dimenzija oslanjajući se na samu strukturu polaznog skupa podataka pri tome otklanjajući za rezultat nebitnu informaciju. Ovakvi algoritmi su nenadzirani, a neki od popularnijih su **PCA** (engl. *principal component analysis*), **MDS** (engl. *multidimensional scaling*) i **LDA** (engl. *linear discriminant analysis*)

2.3.10. Hibridni algoritmi

Kao i kod učenja s više modela hibridno učenje se temelji na više modela koji rješavaju isti problem. Za razliku od "**ensamble learning**"-a koji se temelji na **homogenim** modelima (modelima iste kategorije/vrste) hibridno se učenje radi s **heterogenom** skupinom modela (npr. grupiranje i regresija). Ovime se izbjegava komplikiranost odabira skupine homogenih modela koji, kako se pokazalo, nerijetko ishoduju rezultatima koji imaju loše karakteristike te odluke koje su teške za tumačenje.

2.4. Strojno učenje u medicini

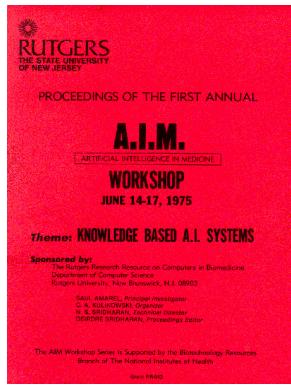
Kao i u drugim disciplinama razvoj primjene umjetne inteligencije i samog strojnog učenja uveliko je bio ograničen nedorašlošću hardvera do početka 21. stoljeća. U posljednjih nekoliko desetljeća bilježi se nagli i vrlo optimističan rast primjene **umjetne inteligencije u medicini** (engl. *AIM - artificial intelligence in medicine*) od personaliziranog pristupa pacijentu, dijagnoze bolesti i predviđanja reakcije na tretmane liječenja istih. Najveći pomaci napravljeni su u području povećavanja obujma pružanja skrbi ubrzanjem postupka sugestivnom dijagnostikom te praćenjem stanja pacijenata što je uvelike poboljšalo izglede oboljelih za priželjkivanim ishodom.



Slika 2.13: Vremenska lenta razvoja umjetne inteligencije u medicini. [5]

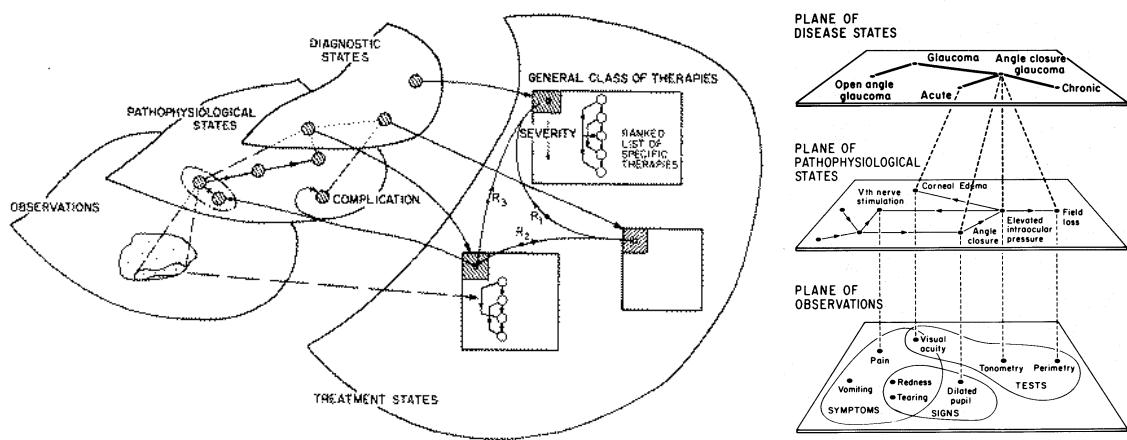
Sami začetci informacijskih sustava vezanih uz medicinu 60.-tih godina prošloga stoljeća strogo su vezani uz Sjedinjene Američke Države. Kao temelj za razvoj naprednih sustava koji se koriste danas imamo sustave za stručnu literaturu (PubMed web pretraživač i Medical Literature Analysis and Retrieval System) koje je Nacionalne knjižnica SAD-a implementirala 1960.-tih kao i u raznim bazama kliničkih podataka i registara povijesti bolesti pacijenata koji se počinju implementirati tih godina.

U razdoblju od 70.-ih do 2000.-ih nastupa prva **zima umjetne inteligencije**, u kojem su ulaganja u razvoj ovog područja manjkava poradi nedostatka interesa kao i visokih troškova održavanja i razvoja digitalnih baza znanja. Upravo u tim uvjetima na američkim sveučilištima Rutgers i Standford počinje razvoj ovog područja osnutkom **programa za istraživanje primjene računalna u bio-medicini** (engl. *The Research Resource on Computers in Biomedicine*) 1971. na Rutgersu te dijeljenog sustava računala za **eksperimente primjene umjetne inteligencije u medicini** (engl. *Medical Experimental–Artificial Intelligence in Medicine*) na Standfordu 1973. Zanimanje u ovo područje potvrdilo je američko ministarstvo zdravljia tako što je 1975. sponsoriralo prvu radionicu u ovom području na sveučilištu Rutgers.



Slika 2.14: Letak prve radionice o umjetnoj inteligenciji u medicini sponzoriranog od američke vlade (slika preuzeta sa Rutgers University⁸).

Prvi primjenjivi model koji je također rezultat rada stručnjaka na Rutgersu je **CASNET** koji je bio u stanju primijeniti znanje o pojedinoj bolesti na pacijenta te liječnicima sugerirati daljnje postupke, rad je predstavljen 1976. na akademiji za oftalmologiju u Las Vegasu.



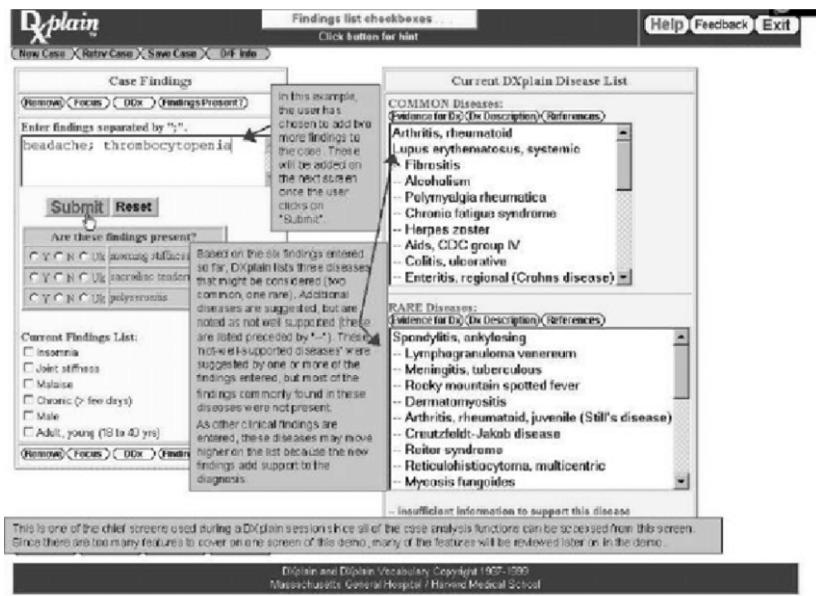
Slika 2.15: Prikaz strukture CASNET modela i troslojnog opisivanja bolesti na primjeru glaukoma (slika preuzeta sa MIT⁹).

Istovremeno je razvijen i **MYCIN** - sustav koji je na osnovu unosa liječnika i šeststotinjak pravila sugerirao lječniku listu mogućih bakterijskih pategona koji uzrokuju opisanu bolest te prigodne antibiotike za tjelesnu težinu pacijenta i bakteriju. Ovaj sustav je bitan jer se iz njega razvio **EMYCIN** kao razvojni okvir za implementaciju sustava temeljenih na pravilima kao što je prvi sustav za dijagnozu u primarnoj medicini **INTERNALIST1**.

⁸Rutgers University: Events, s Interneta: <https://media.cs.rutgers.edu/Events/Amarel/index-slides.html>, 11.09.2023.

⁹MIT: Representation of Expert Knowledge for Consultation: The CASNET and EXPERT Projects, s Interneta: <https://groups.csail.mit.edu/medg/ftp/psz/AIM82/ch2.html>, 11.09.2023.

Osamdesetih godina 209. stoljeća nastupa drugo razdoblje g interesa za ovo područje, od bitnih pomaka vrijedi istaknuti **DXplain** sveučilišta u Massachusettsu koji je za ulaz uzimao opis simptoma te dijagnosticirao pacijenta te služio kao knjižnica mogućih dijagnoza i njihovih detaljnih opisa čiji je originalni broj od 500 naknadno proširen do čak 2400 različitih bolesti.

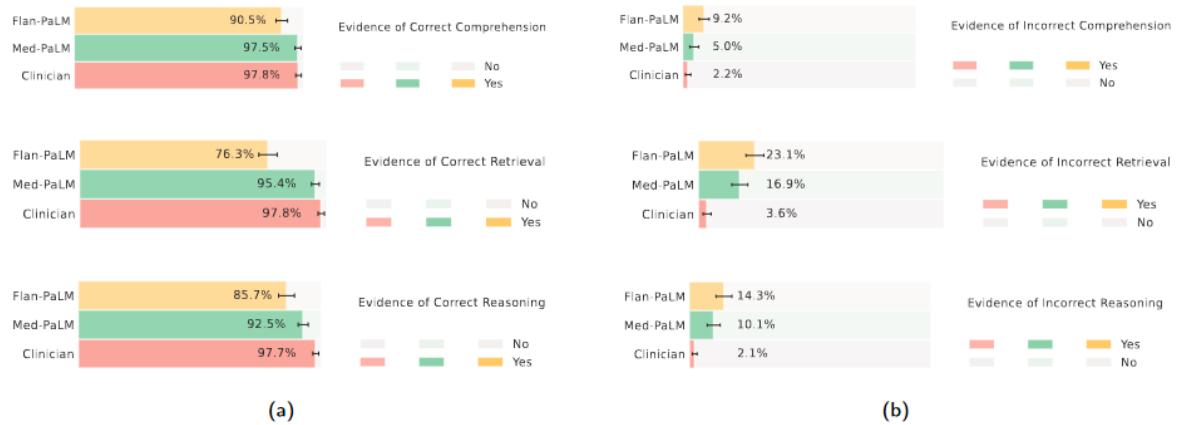


Slika 2.16: Sučelje DXplaina (slika preuzeta sa ResearchGate¹⁰).

2007. godine počinje moderno razdoblje umjetne inteligencije u medicini. Te godine IBM je predstavio sustav za odgovaranje **Watson** temeljen na tehnologiji **DeepQA** koja je za razliku od dosadašnjih sustava **unatražnog zaključivanja** (engl. *backwards chaining*) i sustava sa 'if-then' grananjem, predstavio **zaključivanje unaprijed** (od podataka ka zaključku prateći pravila) - primjenjujući obradu prirodnog jezika i razna pretraživanja nad nestrukturiranim podacima kako bi se pružio odgovor. Primjena ove tehnologije nad bazama znanja iz područja medicine te poviješću bolesti rezultira naprednim sustavom za medicinsko zaključivanje. U razdoblju koje slijedi ovim otkrićima u području primjene umjetne inteligencije na domenu medicine smo vidjeli gotovo sve, od chatbota poput **Mandy iz 2017.** koji služi zaprimanju pacijenata u ordinacijama primarne skrbi, preko dubokih neuronskih mreža i konvolucijskih mreža za slikovnu dijagnostiku pacijenata do jezičnih modela poput **Google-ovog Med-PALM-a** koji na standardiziranim testovima unatrag nekoliko godina prati stvarne liječnike.

¹⁰ResearchGate: A screen from a session demonstrating DXplain, s Interneta: https://www.researchgate.net/figure/A-screen-from-a-session-demonstrating-the-DXplain-diagnostic-system-to-new-users-The_fig2_226706299, 11.09.2023.

¹¹TowardsDataScience: Google Med-PaLM: The AI Clinician, s Interneta: <https://towardsdatascience.com/google-med-palm-the-ai-clinician-a4482143d60e>, 11.09.2023.



Slika 2.17: Rezultati Med-PALM-a na testu koji se sastoji od nasumično izabranih 100 pitanja iz HealthSearchQA, 20 iz LiveQA te 20 iz MedicationQAA (popularni skupovi podataka sa pitanjima iz kliničke primjene). Za ljudski dio eksperimenta uspostavljen je panel kliničkih lječnika koji su dali stručne odgovore na pitanja (slika preuzeta sa TowardsDataScience¹¹).

3. ISTRAŽIVANJE

Ovaj diplomski rad nastao je nad podacima sakupljenim u okviru istraživanja **Referentnog centar Ministarstva zdravstva za medicinu spavanja** pri Kliničko Bolničkom Centru Split. Liječnici istraživači splitskog **Centra za medicinu spavanja** kroz Laboratorij za poremećaje spavanja provode istraživanje o detekciji i proučavanju posljedica apneje i hipopneje kao poremećaja disanja tokom sna. Cilj je dakle olakšati **detekciju** te uvidjeti koji su to faktori povezani sa pojavom ovih poremećaja te koji su mogući **komorbiditeti** istih. Sami dio detekcije te računalnog zaključivanja o potencijalnim komorbiditetima izvodi Fakultet elektrotehnike, strojarstva i brodogradnje u Splitu. Samo istraživanje provodi se nad podacima koji su skupljani kroz preglede pacijenata u laboratoriju u razdoblju **od 2004. do danas**.

3.1. Apneja spavanja

Apneja (grk. $\alpha\pi\nu\omega$ - manjak disanja). spavanja je kliničko stanje kod kojeg pacijent pati od **prestanaka disanja** tokom sna. Apneja se pojavljuje u dva oblika - **obstruktivna apneja (OSAHS)** kod koje dolazi do začepljenja gornjih dišnih puteva i **centralna apneja (CSA)** kod koje dolazi do neispravne kontrole disanja od strane mozga. Ovakav prestanak disanja izaziva refleks koji budi pacijenta no time ometa spavanje. [9]

3.1.1. Klasifikacija ozbiljnosti oboljenja apneje

Severitet poremećaja računa se pomoću **indeksa apneje i hipopneje - AHI** (engl. *apnea hypopnea index*) koji predstavlja broj pojava apneje (kada osoba nema protok zraka na više od 10 sekundi) ili hipopneje (kada je protok zraka smanjen za barem 30% na više od 10 sekundi uz buđenje i desaturaciju krvi kisikom).

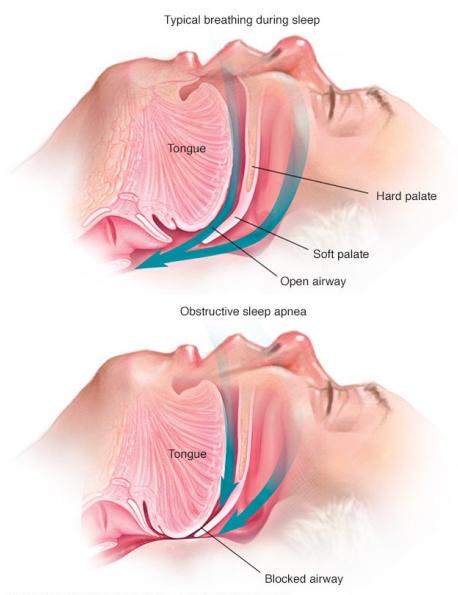
AHI	Ozbiljnost oboljenja
≤ 5	zdrava osoba
≥ 5	blaga apenja
≥ 15	umjerena apneja
≥ 30	jako apneja

Tablica 3.1: Klasifikacija ozbiljnosti oboljenja apneje.

3.1.2. Opstruktivna apnea

Uzroci

Do ovog poremećaja dolazi zbog **opuštanja mišića** stražnjeg dijela grla koji podržava (zateže) meko nepce, nepčanu resicu, krajnike same stjenke grla te jezik. Opuštanjem ovog mišića navedeni dijelovi kanala se spuštaju, zatvaraju kanal što uzrokuje nedostiznu opskrbu kisikom te pad razine istoga u krvi. Primjetivši nedostatak kisika, mozak pokreće **refleks buđenja** kako bi se raširio dišni put, ovakav refleks je toliko kratak da ga se po buđenju pacijent ne sjeća.



Slika 3.1: Prestanak disanja uslijed zatvaranja dišnih puteva. [8]

Faktori povećanja pojavnosti

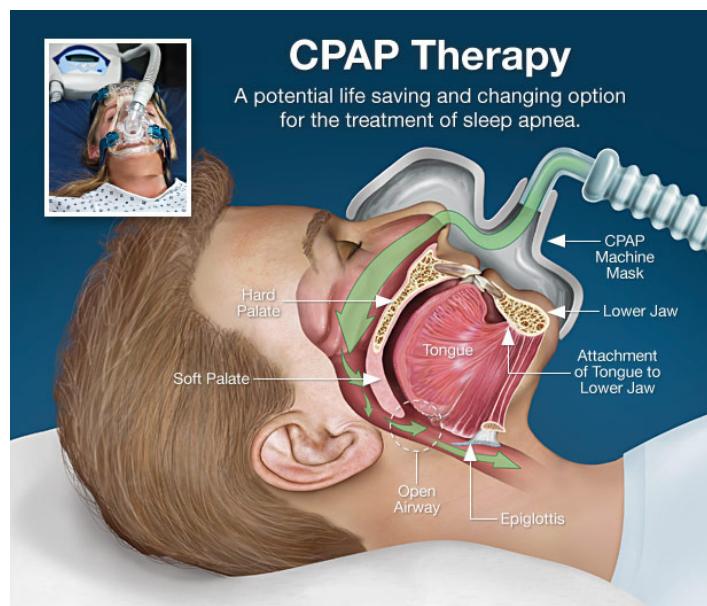
Od opstruktivne apneje znatno obolijevaju **pretile osobe** zbog masnih naslaga oko dišnog puta, također su posebno sklone bolesti **osobe sa specifičnom kraniofacijalnom anatomijom** (širokim vratom) s obzirom da one nerijetko imaju uže dišne puteve (mikrognatija). Također bilo kakva zadebljanja tkiva dišnih puteva poput krajnika, raznih polipa povećavaju vjerojatnost za pojavu apneje. Također će vjerojatnost biti povećana za osobe koje konzumiraju **alkohol** ili bilo kakve **lijekove za smirenje** s obzirom da i u tom slučaju dolazi do opuštanja mišića. Osim ovih predispozicija s povećanim rizikom apneje povezane su razne dijagnoze poput hipertenzije, dijabetesa, astme, alergija koje utječu na začepljenošću nosa, hormonalnih poremećaja te osoba koje su patile od moždanog udara ili raka pluća. [9]

Komorbiditeti

Kao posljedica ovakvog isprekidanog sna nastaje **iscrpljenost**, konstantni umor i razdražljivost. Pored iscrpljenosti, nagle oscilacije u zasićenosti krv i kisikom ostavljaju tragove na kardiovaskularni sustav povišujući rizik za razvijanje **hipertenzije** (povećanog krvnog tlaka), aritmije te srčanog udara. Također su vidljive posljedice po jetru, gdje se pojavljuju ožiljci, a taj se poremećaj naziva **nealkoholičarskom bolešću masne jetre**. [8]

Liječenje

Osim otklanjanja uzročnika popu prekomjerne tjelesne težine, alkohola i sedativa, OSHAS se liječi primjenom **kontinuiranog pozitivnog zračnog pritiska - CPAP** (engl. *continuous positive airway pressure*). Ta se terapija provodi postavljanjem nazalno-oralne maske preko koje se dovodi kisik pri određenom zračnom pritisku prilagođenom dišnim putevima pacijenta.[8]



Slika 3.2: Liječenje primjenom kontinuiranog pozitivnog zračnog pritiska (slika preuzeta sa Perth CPAP¹).

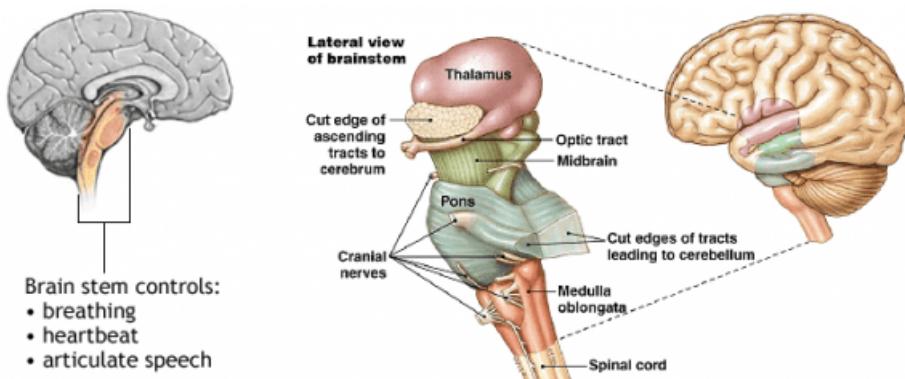
Osim CPAP-a kod blažih pojava apneje bolest se može tretirati udlagama za reponiranje mandibule kojima se mijenja zagriz pacijenta tako da se zbacuje čeljust time proširuju dišni putevi ždrijela. Postoje i operacijska rješenja za tretiranje OSHAS-a poput uklanjanja krajnika, operacija čeljusti te traheostoma (prolaz iz dušnika ka vanjskom dijelu vrata čime se zaobilaze gornji dišni putevi) koja je jako dobro, ali rijetko korišteno rješenje. [9]

¹Perth CPAP: What is CPAP?, s Interneta: <https://www.perthcpap.com.au/blogs/news/what-is-cpap>, 11.09.2023.

3.1.3. Centralna apneja

Uzroci

Centralna apneja uzrokovana je **ne slanjem živčanih impulsa ka mišićima koji pokreću dišanje** od strane mozga. Ovakvi napadaji popraćeni su buđenjem, zadihanjušću i nemogućnošću ponovnog sna.



Slika 3.3: Dio mozga koji upravlja podsvjesnim radnjama (slika preuzeta sa Quizlet²).

Faktori povećanja pojavnosti

Osim starijih muškaraca i pretilih osoba za razvoj poremećaja centralne apneje povećan rizik imaju i osobe koje su preživjele moždani udar, osobe koje pate od hipoksije (npr. velike visine) te osobe koje su koristile ili koriste opijate ili lijekove protiv bolova. [8]

Komorbiditeti

S obzirom na prirodu ovih napadaja i nemogućnost sna, centralna apneja ima jako izraziti umor za svoju najjasniju posljedicu. Osim umora uzrokuje i poremećaje kardiovaskularnog sustava uslijed pojave epizoda niske zasićenosti kisikom.

Liječenje

Centralna apneja se osim uklanjanjem uzročnika može tretirati i **adaptivnom servoventilacijom** (jako slično samom CPAP-u) koja omogućava potporu u disanju tokom apneja i hipopneja.

²Quizlet: Brainstem and Brainstem Lesions, s Interneta: <https://quizlet.com/582291864/brainstem-and-brainstem-lesions-flash-cards/?setIdOrUsername=582291864>, 11.09.2023.

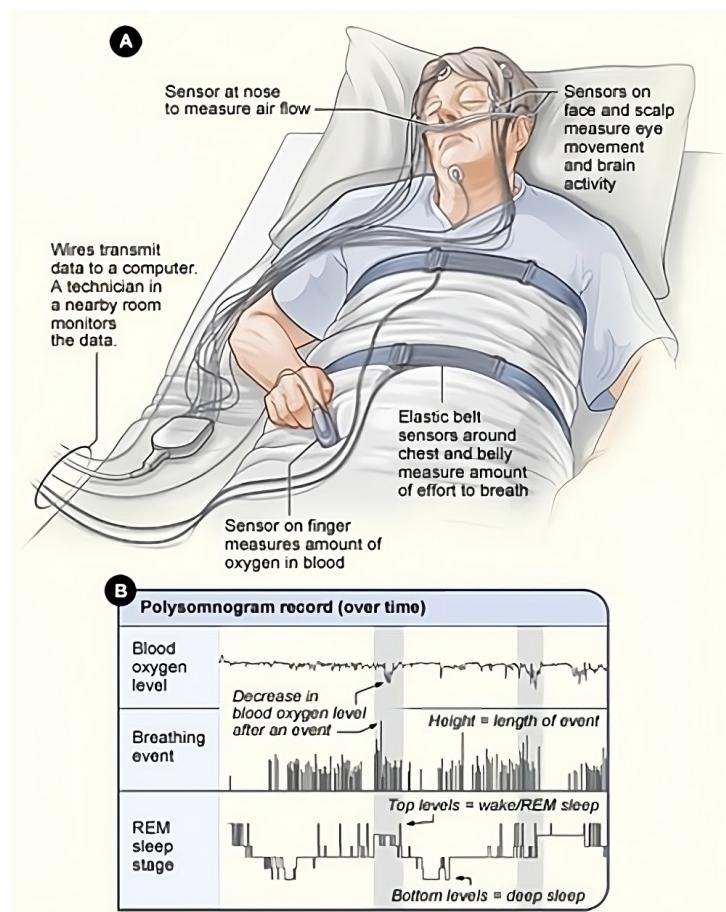
3.2. Polisomnografija

Polisomnografija je neinvazivna stručna studija o snu osobe, radi se kako bi se dijagnosticirali poremećaji spavanja poput apneje. Sama studija temeljena je na proučavanju korelacije između raznih testova koji se vode tokom noći sna i fazi sna mozga promatranika. Naime u normalnom ciklusu san započinje ne-REM fazama, u kojima aktivnost mozga slabiti, te nakon otprilike sat vremena mozak ulazi u **REM fazu (engl. rapid eye movement)** koja je praćena povećanom moždanom aktivnošću (pomicanje očiju i većina produkcije snova je u ovoj fazi). Ovakve izmjene se događaju **ciklički** svako otprilike 90 minuta te je poremećaj u izmjeni istih uzrokovani bolestima spavanja. [10]

Tokom noći provode se različiti testovi:

Signal	Cilj mjerjenja
EEG - Elektroencefalogram	Mjeri aktivnost moždanih valova.
EOG - Elektrookulogram	Snima pokrete očiju.
EMG - Elektromiogram	Mjeri aktivnost skeletnih mišića.
EKG - Elektrokardiogram	Snima brzinu i ritam srca.
Senzor za protok zraka kroz nos	Mjeri protok zraka kroz nos.
Senzor za protok zraka kroz usta	Mjeri protok zraka kroz usta.
Torakalne respiratorne trake	Mjeri pokrete prsnog koša.
Abdominalne respiratorne trake	Mjeri pokrete trbuha.
Mikrofon	Snima aktivnost hrkanja.
Pulsna oksimetrija	Mjeri razine kisika u krvi.
Kapnografija	Mjeri razine ugljikovog dioksida.
PLETH - Fotopletizmogram	Snima promjene u volumenu krvi u organu.
Temperaturni senzori	Prate temperaturu kože ili tijela.
Senzori za položaj tijela	Otkrivaju promjene u položaju tijela.
GSR - Galvanski odziv kože	Mjeri provodljivost kože.
Aktigrafija nogu	Snima pokrete nogu.

Tablica 3.2: Signali polisomnografije.



Slika 3.4: Ilustracija pacijenta tokom polisomnografije te analize nastalih signala (slika preuzeta sa PsychDb³).

Sobe u kojima se obavlja polisomnografsko istraživanje opremljene su kamerama za rad u niskom svjetlu te audio sustavima kojima tehničari mogu komunicirati sa samim pacijentom. Rezultat ovakvog istraživanja su najčešće razni indeksi poput AHI i ODI koji govore o broju prestanaka disanja većih od 10 sekundi te značajnih padova u saturaciji krvi kisikom te u slučaju povećanih vrijednosti upućuju na poremećaje sna poput apneje.

³PsychDB: Polysomnography (PSG), s Interneta: <https://www.psychdb.com/neurology/polysomnography>, 11.09.2023.

4. PODACI

4.1. Prikupljanje podataka

Proces prikupljanja podataka odvijao se kroz više posjeta **Centru za medicinu spavanja Kliničko Bolničkog Centra Split**. Tamo je Fakultetu elektrotehnike, strojarstva i brodogradnje dan pristup CD (engl. *compact disc*) zapisima s polisomnografijama pacijenata te lista šifri pacijenata sa teškim oblicima apneje. Ovi podaci su u Centru prikupljeni od 2004. godine te prikupljanje još uvijek traje. Centar ima dvije sobe za polisomnografiju opremljene kamerama i audio sustavom za motrenje pacijenta tokom noći te komunikaciju s istim. Za samo prikupljanje u Centru se koristi uređaj **Philips Alice PDX** opisan u idućem paragrafu.

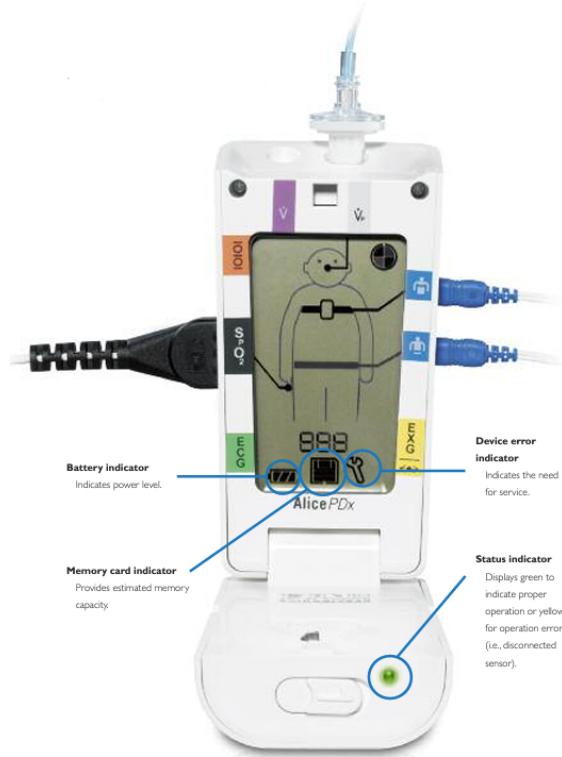
4.1.1. Philips Alice PDX

Alice PDX je prijenosni dijagnostički snimač sesija spavanja namijenjen za pretrage vezane uz opstruktivnu apneju te općenitu u u proučavanju poremećaja spavanja. Sam uređaj krasiti velika jednostavnost korištenja sa bojama označenim priključcima i slikovitim uputama za spajanje raznih senzora što ga čini jako prikladnim za samostalnu kućnu upotrebu. Signali se snimaju na prijenosnu memorijsku karticu ili računalo koje je spojeno na Alice te na sebi ima pokrenut **Alice Sleepware** aplikaciju. Sleepware je aplikacija koja omogućuje udaljenu automatsku analizu snimljenih podataka o snu, pritom mu nije cilj zamijeniti stručnjake. Također, Sleepware pokriva potrebu za dodatnim senzorima koji su pacijentu potrebni za dijagnostiku i sjedinjuje ih u jedan skup podataka. [7]



Slika 4.1: Korištenje Alice PDx snimača. [7]

Uređaj dolazi sa 1GB pohrane u obliku SD kartice, prsnim nosačem za uređaj, 2 Pro-Tech zRIP pojasa za mjerjenje naprezanja prsnog koša, termalnim senzorom koji se postavlja u oralnu šupljinu, Nonin SpO_2 senzorom, senzorom za tok zraka, senzorima za poziciju tijela i otkucaje srca. Van osnovne konfiguracije moguće je postaviti dodatnih 7 kanala (3 glavna i 4 izvedena) za EKG, 4 kanal za EEG ili EOG te 3 kanala za EMG.



Slika 4.2: Alice PDx [7]

Sam uređaj ima bateriju koja je u stanju snimiti 2-3 sesije spavanja te je moguće automatsko pokretanje i zaustavljanje sesije kako bi se otklonila ljudska greška. Uređaj je težak 230 grama i veličine 12.7cm x 7.62cm x 5.08cm. Razlučivost signala donekle je ograničena brzi-

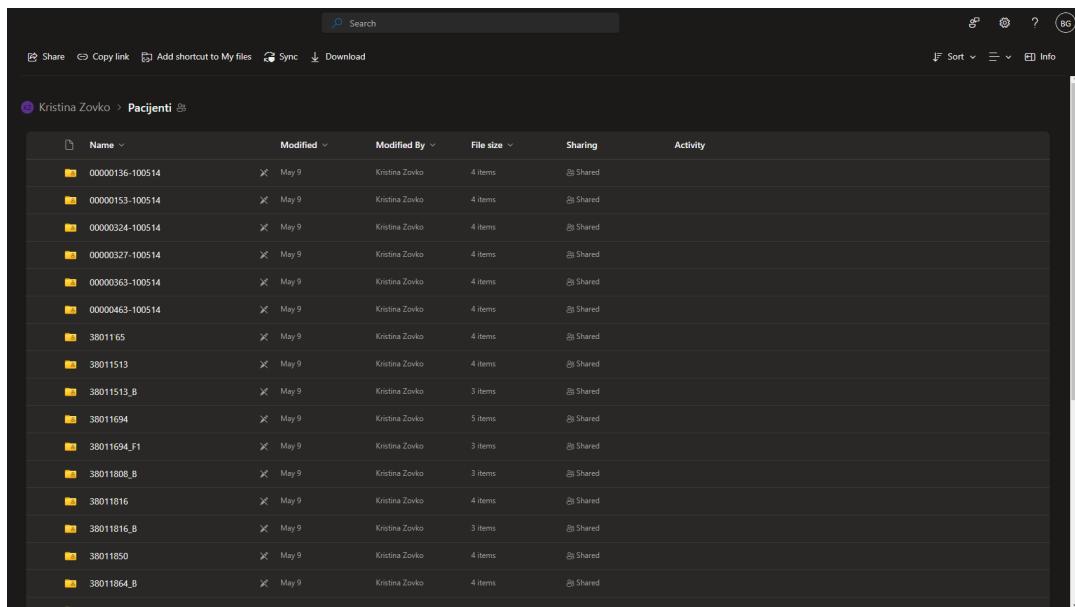
nom pohrane s toga su signali ograničeni na 200Hz za ExG signale, 100Hz za termalne senzore i senzore naprezanja prsnog koša, 200Hz za senzore protoka zraka, 1Hz za poziciju tijela te 500Hz za senzore za detekciju hrkanja. Same numeričke vrijednosti uzorka precizne su do 16 bita.

	Impedancija ulaza	Bandwidth	Raspon ulaznog signala
ECG	10 MΩ po elektrodi 20 MΩ diferencijal	0.318 to 81 Hz	b ± 4 mV
EEG	2 MΩ po elektrodi 4 MΩ diferencijal	0.318 to 35 Hz	c ± 500 μV
EMG	2 MΩ po elektrodi 4 MΩ diferencijal	9.7 to 86 Hz	d ± 150 μV

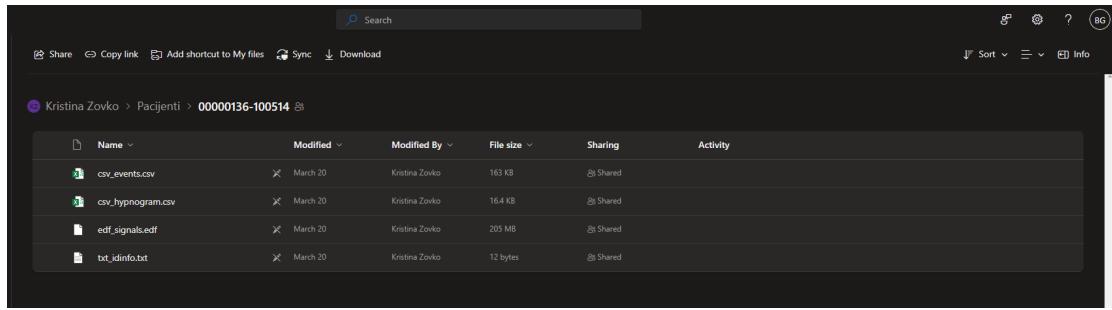
Tablica 4.1: Karakteristika ulaza za ExG signale.

4.2. Obrada i interpretacija

Podaci koji su prikupljeni u Centru za poremećaje spavanja postavljeni su na zajednički disk u oblaku na platformi OneDrive. U repozitoriju su mape imenovane šifrom pacijenta radi zaštite podataka. Svaka mapa sadrži tri signala, *csv_events.csv*, *csv_hypnogram.csv* i *edf_signals.edf*.



Slika 4.3: OneDrive sharepoint sa polisomnografijsama pacijenata.



Slika 4.4: Podaci pojedinog pacijenta.

Events signal je "event driven" signal, tj. ne bilježi uzorke kontinuirano, već zapisi u tablici sadrže događaje, vremensku oznaku kad su se dogodili i koliko su trajali. Osim **tipa događaja u snu** (*opstruktivna apneja, centralna apneja, miješana apneja, hipopneja, pokret nogu, desaturacija krv u kisikom, trahikardija, bradikardija, hrkanje* te odziv raznih senzora.), zapisi sadrže **vremensku oznaku, trajanje u sekundama, epohu** (jedna epoha duga je 30 sekundi), **samu fazu sna** (WK, N1, N2, N3, REM), **položaj tijela** (Up, Prone, Supine, Left, Right) i oblik validacije zapisa.

```

[4]: import pandas as pd

[17]: root_path="C:\\\\users\\\\bruno\\\\Desktop\\\\Diplomski\\\\38011929\\\\"
edf_path = root_path + "edf_signals.edf"
hypnogram_path = root_path + "csv_hypnogram.csv"
events_path = root_path + "csv_events.csv"

[18]: hypo_df = pd.read_csv(hypnogram_path)
events_df = pd.read_csv(events_path, sep=';')

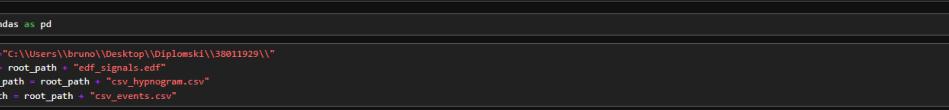
[19]: events_df.head()

[19]:
   Type Stage Time Epoch Duration Body Position Validation
0  Gain (CEMG)  WK  21:45:47    1    0,0      Up  Automatic Gain
1  Gain (CanFlow)  WK  21:45:47    1    0,0      Up  Automatic Gain
2  Gain (TFlow)  WK  21:45:47    1    0,0      Up  Automatic Gain
3  Gain (THO)  WK  21:45:47    1    0,0      Up  Automatic Gain
4  Gain (ABD)  WK  21:45:47    1    0,0      Up  Automatic Gain

```

Slika 4.5: *events_csv* zapis pojedinog pacijenta.

Hypnogram signal je "time driven" signal koji kontinuirano svako 30 sekundi bilježi **fazu sna - WK (budan), N1, N2, N3 i REM**.



```
[4]: import pandas as pd

[17]: root_path="C:\\Users\\bruno\\Desktop\\Diplomski\\38011929\\"
       edf_path = root_path + "edf_signals.edf"
       hypnogram_path = root_path + "csv_hypnogram.csv"
       events_path = root_path + "csv_events.csv"

[18]: hypno_df = pd.read_csv(hypnogram_path)
       events_df = pd.read_csv(events_path, sep=';')

[25]: hypno_df.head()

[25]: Epoch Number Start Time Sleep Stage
      0           1  21:45:47      WK
      1           2  21:46:17      WK
      2           3  21:46:47      WK
      3           4  21:47:17      WK
      4           5  21:47:47      WK

[31]: hypno_df['Sleep Stage'].unique()
[31]: array(['WK', 'N1', 'N2', 'N3', 'REM'], dtype=object)
```

Slika 4.6: *hypnogram_csv* zapis pojedinog pacijenta.

Treći i informacijama najgušći zapis je signal u **EDF** (engl. *euopean data format* formatu koji sadrži vrijednosti svih senzora uzorkovane (time driven) svakih 2 milisekunde (frekvencija od 500Hz).



File Edit View Run Kernel Tabs Settings Help

SL23-SeeSignals.ipynb

Notebook Python 3 (ipykernel)

```
[42]: import pandas as pd
import mne

[43]: root_path = "C:\Users\bruno\Desktop\Diplomski\J8011929\"
edf_path = root_path + "edf_signals.edf"

[*]: data = mne.io.read_raw_edf(edf_path, preload=True)

data_raw = data.get_data()
data.info = data.info
data_channels = data.ch_names

Extracting EDF parameters from C:\Users\bruno\Desktop\Diplomski\J8011929\edf_signals.edf...
EDF file successfully read.
Setting channel info structure...
Creating raw.info structure...
Reading raw ... 151309999 - 0.000 ... 30261.998 secs...
C:\Users\bruno\Appldata\Local\Temp\ipykernel_4664\110522725.py:1: RuntimeWarning: Channel names are not unique, found duplicates for: {'Flow Patient'}. Applying running numbers for duplicates.
  data = mne.io.read_raw_edf(edf_path, preload=True)

[*]: signals = pd.DataFrame(data_raw)
signals.signal
signals.columns = data_channels

[*]: pd.set_option('display.max_columns', None)

[40]: signals.head()
```

	EOG LOC A2	EOG ROC A2	EEG F3-A2	EEG F4-A1	EEG A1-A2	EEG C3-A2	EEG C4-A1	EEG O1-A2	EEG O2-A1	EMG Chin	Flow Patient 0	Flow Patient 1	Effort THO	Effort ABD	Snore	ECG I	SpO2	PulseRate	Leg 1	Leg 2	Body
0	4.776074e-09	1.190204e-09	0.001526	0.001526	0.001526	0.001526	1.271534e-07	0.0	0.0	1.190204e-09	1.190204e-09	0.0									
1	1.611297e-05	-4.420119e-06	0.0494376	8.511880e-06	4.436046e-06	-4.679344e-06	5.867698e-06	-3.338348e-06	-6.416991e-06	-5.541646e-06	-0.700382	0.965666	5.731984	2.711246	0.001526	1.464620e-04	0.0	0.0	-2.989307e-06	-8.802543e-07	0.0
2	7.551205e-05	-1.995248e-06	1.578718e-06	3.973859e-06	2.112405e-06	-2.129832e-06	2.648085e-06	-1.614176e-06	-3.404880e-07	-3.008597e-06	-0.945230	1.302066	7.729230	3.655733	0.001526	4.151673e-05	0.0	0.0	-1.735212e-06	-4.838715e-07	0.0
3	-6.069541e-06	1.561890e-06	-1.359795e-06	-3.151915e-06	-1.742184e-06	1.655032e-06	-2.050012e-06	1.349838e-06	3.148307e-06	2.733477e-06	-0.807776	1.113310	6.606129	3.124704	0.001526	-1.312850e-05	0.0	0.0	1.653121e-06	4.443866e-07	0.0
4	-8.223518e-06	2.052019e-06	1.052196e-06	-4.225986e-06	-2.384823e-06	2.147040e-06	-2.692824e-06	1.882866e-06	4.664249e-07	4.091086e-06	-0.434347	0.600341	3.558127	1.683394	0.001526	7.196793e-06	0.0	0.0	2.563963e-06	6.675325e-07	0.0

Slika 4.7: *edf_signals* zapis pojedinog pacijenta.

4.2.1. Formati pohrane

S obzirom da učitavanje podataka o polisomnografiji dvadesetak pacijenata oduzima podosta vremena - svaki zapis u EDF formatu je veličine oko 2GB. Potrebno je bilo sagledati koji format je prikladniji za rad.

EDF - European Data Format

EDF je jednostavan fleksibilan format za pohranu i razmjenu više-kanalnih signala primjenjivan u sferi bio-medicine. Razvijen je od strane skupine europskih liječnika i inženjera koji su prvi sastanak imali na *Sleep Congress Copenhagen* konferenciji 1987. godine. Pod vodstvom profesorice Annelise Rosenfalck sa sveučilišta u Aalborgu pokrenut je projekt **Metodologija za analizu kontinuuma sna i jave** (engl. *Methodology for the Analysis of the Sleep-Wakefulness Continuum*), projekt je podržan od strane europske udruge COMAC (fr. *Comité d'Action Concertée*) za biomedicinsko inženjerstvo i trajao je od 1989. do 1992.

Sam format je objavljen 1992. godine u 82. izdanju časopisa **Electroencephalography and Clinical Neurophysiology** i od tada je smatran standardnim formatom za snimke EEG-a i polisomnografije. Sam zapis se sastoji od 2 vrste zaglavlja, prvo zaglavljje je samo 1 i sadrži podatke o ukupnom zapisu, dok druga zaglavljva opisuju pripadajuće signale zapisa. Same vrijednosti uzoraka predstavljene su kao 2-bajtni cijeli brojevi u obliku dvojnog komplementa. [11]

Veličina unosa	Značenje
8 ASCII	Verzija formata (0)
80 ASCII	Lokalna identifikacija pacijenta
80 ASCII	Lokalna identifikacija snimanja
8 ASCII	Datum početka snimanja (dd.mm.yy)
8 ASCII	Vrijeme početka snimanja (hh.mm.ss)
8 ASCII	Broj bajtova u zaglavljju
44 ASCII	Rezervirano
8 ASCII	Broj podatkovnih zapisa (-1 ako je nepoznato)
8 ASCII	Trajanje podatkovnog zapisa (u sekundama)
4 ASCII	Broj signala (ns) u podatkovnom zapisu

Tablica 4.2: Zapisи prvog EDF zaglavlja.

Kao što je vidljivo na slici 4.8 ovaj format pruža izvrsnu kompresiju sa 15-ak milijuna zapisa pohranjenih u datoteku od oko 2GB. Problem formata su duga vremena čitanja i pisanja i komplikacije pri pretvorbi (postavljanje zaglavlja) između njega i okvira podataka u Pandas-u. Za manipulaciju ovim datotekama koristi se MNE - Python paket za istraživanje, vizualizaciju i analizu neurofizioloških podataka kod ljudi (podrška za MEG, EEG, sEEG, ECoG, NIRS i ostala mjerena).

Veličina unosa	Značenje
ns * 16 ASCII	ns * Oznaka
ns * 80 ASCII	ns * Tip pretvornika
ns * 8 ASCII	ns * Fizička dimenzija
ns * 8 ASCII	ns * Fizički minimum
ns * 8 ASCII	ns * Fizički maksimum
ns * 8 ASCII	ns * Digitalni minimum
ns * 8 ASCII	ns * Digitalni maksimum
ns * 80 ASCII	ns * Prethodno filtriranje
ns * 8 ASCII	ns * Br. uzoraka u svakom podatkovnom zapisu
ns * 32 ASCII	ns * Rezervirano

Tablica 4.3: Zapisi drugih EDF zaglavlja. [11]

Vrijednost	Parametar
nr of samples[1] * integer	Prvi signal u podatkovnom zapisu
nr of samples[2] * integer	Drugi signal
...	...
nr of samples[ns] * integer	Posljednji signal

Tablica 4.4: Parametri podatkovnog zapisa u EDF. [11]

CSV - Coma Sepparated Values

CSV je svima poznat format za tabličnu pohranu podataka. Sami format je prvi put podržan 1972. u sklopu IBM-ovog Fortrana. CSV datoteka se sastoji od redaka u kojima su vrijednosti razdvojene razmakom. Prvi redak predstavlja imena stupaca dok ostali redci predstavljaju pojedine unose. Iako je izrazito jednostavan za čitanje i razumijevanje te široko podržan za rad s bibliotekama poput Pandas-a, ovaj format bilježi uvjerljivo najgore performanse u sva tri vida - brzini pisanja i čitanja kao i veličini pohranjenog zapisa na disku kao što je vidljivo na slici 4.8.

Feather

Feather je format koji omogućava **binarnu stupčanu serijalizaciju okvira podataka**. Dizajniran je upravo kako bi se pohrana, razmjena i korištenje okvira podataka u različitim jezicima za analizu podataka učinila bržom i efikasnijom. Originalnu verziju za Python napisao je 2017. godine **Wes McKinney** dok je za R to učinio **Hadley Wickham**. Feather je temeljen na stupčanoj memorijskoj specifikaciji za pohranu binarnih podataka na disku koja je postavljena u okviru **Apache Arrow** projekta. Upravo to čini operacije čitanja i pisanja izvanredno brzima. Također, temeljenje na ovoj specifikaciji omogućava enkodiranje null i NA vrijednosti kao i tipove varijabilne duljine poput UTF8. Iako je format temeljen na Apache Arrowu definira vlastite pojednostavljenje sheme i meta-podatke za pohranu na disku. [12]

Osim što Feather pokazuje ubrzanje reda veličine 10 na operacijama čitanja i pisanja vidljivo na slici 4.8, rad s formatom je veoma jednostavan koristeći Pyarrow biblioteku za Python koja omogućuje unutar-memorijsku podatkovnu analizu, što ga je učinilo krajnjim odabirom za rad.

Podržane vrste stupaca u Featheru

Širok raspon numeričkih tipova (int8, int16, int32, int64, uint8, uint16, uint32, uint64, float, double). Logičke/boolean vrijednosti.

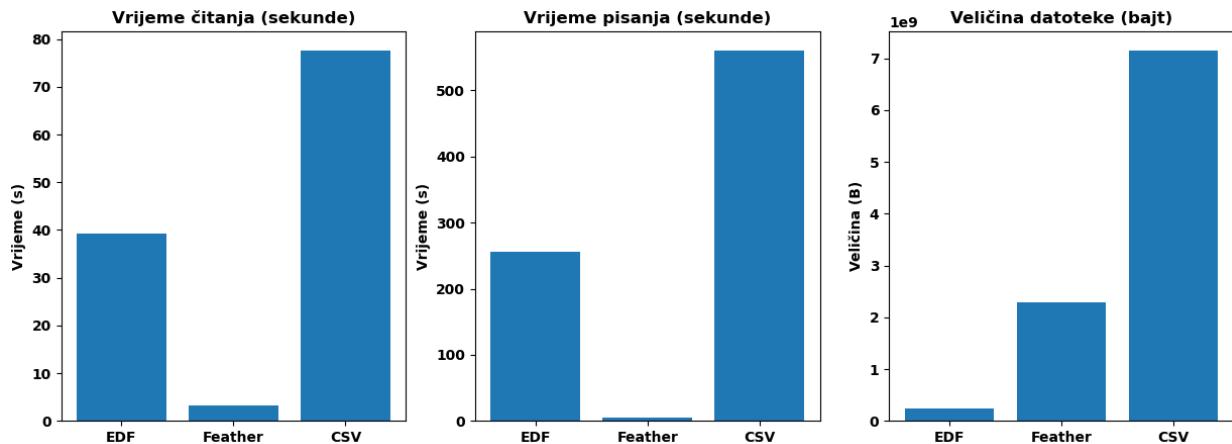
Datumi, vrijeme i vremenske oznake.

Faktori/kategoričke varijable s fiksnim skupom mogućih vrijednosti.

UTF-8 kodirani nizovi.

Proizvoljni binarni podaci.

Tablica 4.5: Podržane vrste stupaca u Featheru. [12]



Slika 4.8: Usporedba različitih formata pohrane podataka o polisomnografiji.

5. OKRUŽENJE

5.1. Anaconda

Anaconda je besplatna platforma koja predstavlja distribuciju za Python i R, upravljač paketima i okruženjima - osnovana je 2012. od strane poduzetnika Petera Wanga kako bi se popratio izuzetan rast primjene analize podataka u poslovnoj sferi. Dolazi sa 250-ak pred-instaliranih paketa za rad u sferama strojnog učenja, podatkovne znanosti i obrade velikih kolčina podataka. Cilj je pojednostavljenje upravljanja okruženjima i paketima. Dolazi sa svojim upraviteljem paketa **Conda** čiji je razvoj imao za cilj nadomjestiti nedostatak konzistentnog upravljanja zavisnim paketima koje je implementirao **pip** - zavisnost jednog paketa o verziji drugog paketa instalirala bi automatski tu verziju bez provjere zavisnosti već instaliranih paketa (**pip** je ovo od tada popravio). Uz to, Anaconda dolazi sa GUI-jem zvanim Anaconda Navigator kako bi se omogućilo lakše korištenje za korisnike koji nisu navikli na korištenje CLI-ja. [13]

5.2. Google Colab

Google Colab je usluga pokretanja Jupyter bilježnica na oblaku napravljena od strane Google Research tima. Usluga kao ciljanu svrhu ima pomoć u zadacima vezanim uz obradu podataka i strojno učenje. Osnovni nivo usluge je besplatan i omogućava korisnicima pokretanje bilježnica pohranjenih na Google Disku koristeći računalne resurse (CPU, GPU, RAM) u oblaku. Glavni problem na koji sam naišao u radu sa ovom uslugom je netransparentnost vezana uz korištenje resursa. Naime, Google ne pruža informacije o resursima koji možete koristiti već radi nekakav oblik dinamične alokacije istih. S obzirom da je objašnjenje za takvo djelovanje upravo povećanje efikasnosti besplatnog korištenja resursa, postavlja se pitanje zašto je ista situacija i sa plaćenim planovima u kojima je nekada dopušteno pokretanje jedne sesije, a nekada tri, pto sustav čini izrazito nepouzdanim. Sami sustav omogućuje odabir između CPU ili GPU okruženja (odabir rada sa A100, V100, T4 ili TPU grafičkim karticama) i običnog ili okruženja sa većom količinom radne memorije (engl. *high RAM*). [14]

5.3. Azure ML

Azure Machine Learning je Microsoftova platforma za strojno učenje koja omogućuje pokretanje virtualnih okruženja za rad s podacima. Podaci se pohranjuju na Azure serverima u **data store**-ove kojima se kasnije pristupa iz Jupyter bilježnica. Platforma je za razliku od Google-ove mjesecne pretplate postavljena kao PaaS (engl. *platform as a service*) sa plaćanjem po satu korištenja, gdje je ovisno o računalnoj moći same virtualne mašine prikazana cijena sata korištenja (od 0\$ do 2\$).

5.4. Jupyter

Jupyter je projekt usredotočen na razvojne alate za interaktivno računarstvo, primarno vezano uz područje podatkovne znanosti. Projekt je najpoznatiji po Jupyter Notebook web aplikaciji koja omogućava kombinaciju računalnog koda kao i teksta te multimedije (engl. *rich text*). [15] Ovi dokumenti su popularni upravo jer kao takvi služe kao izvršivi programski kod za analizu te prikaz rezultata analize i opise u formatu čitljivom čovjeku. Na "backendu" same aplikacije nalazi se **jezgara** (engl. *kernel*) koji alocira računalne resurse računala domaćina i izvršava kod sadržan u bilježnici pisan u odabranom jeziku (svaki jezik različit kernel). [16]

6. IMPLEMENTACIJA

6.1. Automatizacija i optimizacija

Kao prvi zadatak unutar rada postavilo se ubrzanje izračuna ODI indeksa te iscrtavanja podataka vezanih uz saturaciju krvi kisikom. Naime postojeća Jupyter bilježnica pokretala se ručno za pojedinog pacijenta uz prilagođavanje parametara kojima se ciljao dohvati stupca sa uzorcima saturacije iz EDF zapisa - s obzirom da signali nisu uvijek isto poredani, tražio bi se stupac sa vrijednostima koje gravitiraju oko 90 čime se prepoznaje da se radi o signalu saturacije krvi kisikom. Ovaj problem uzrokovani je nezgrapnošću EDF-a, odnosno zanemarivanjem njegovog zaglavlja. Jednostavnim **čitanjem zaglavlja** moguće je pronaći signal pomoću ključa "SpO2".

```
data = mne.io.read_raw_edf(edf_path)

data_raw = data.get_data()
data_info = data.info
data_channels = data.ch_names

saturation_index = data_channels.index('SpO2')
display(saturation_index)

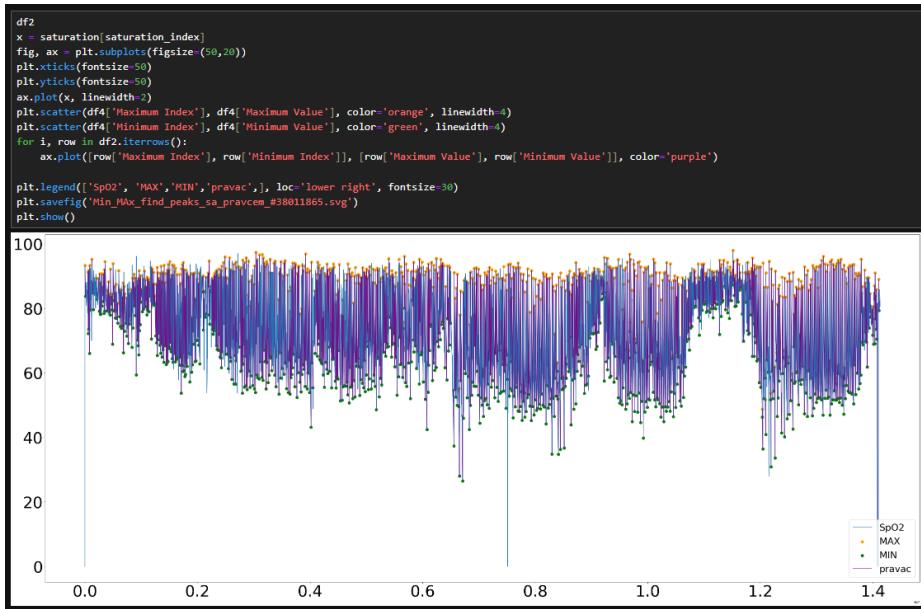
signal = pd.DataFrame(data_raw)

signal_display = signal.T
signal_display.columns = data_channels
signal_display

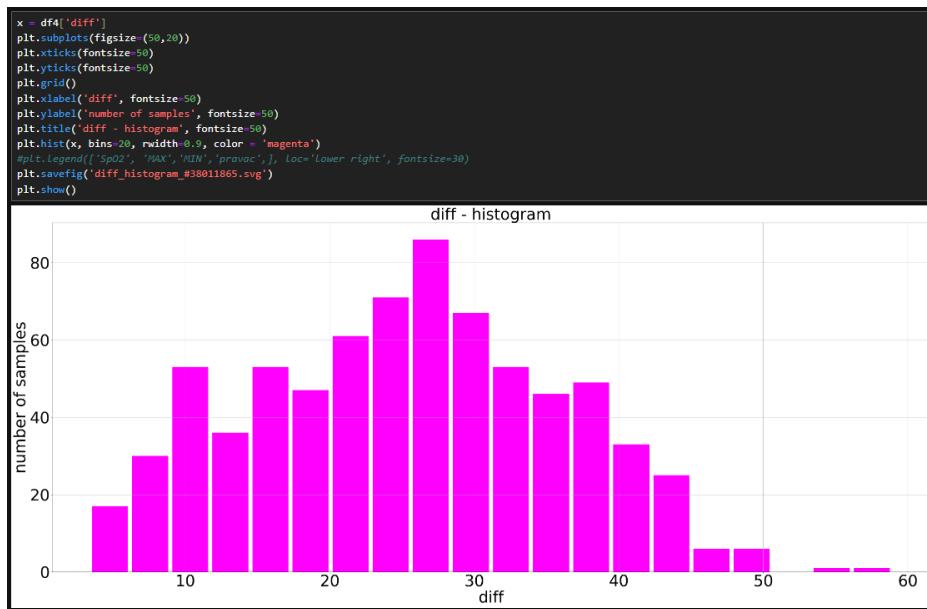
saturation = signal.iloc[saturation_index]
saturation = pd.DataFrame(saturation)
saturation
```

Slika 6.1: Izdvajanje zapisa o zasićenosti krvi iz EDF datoteke.

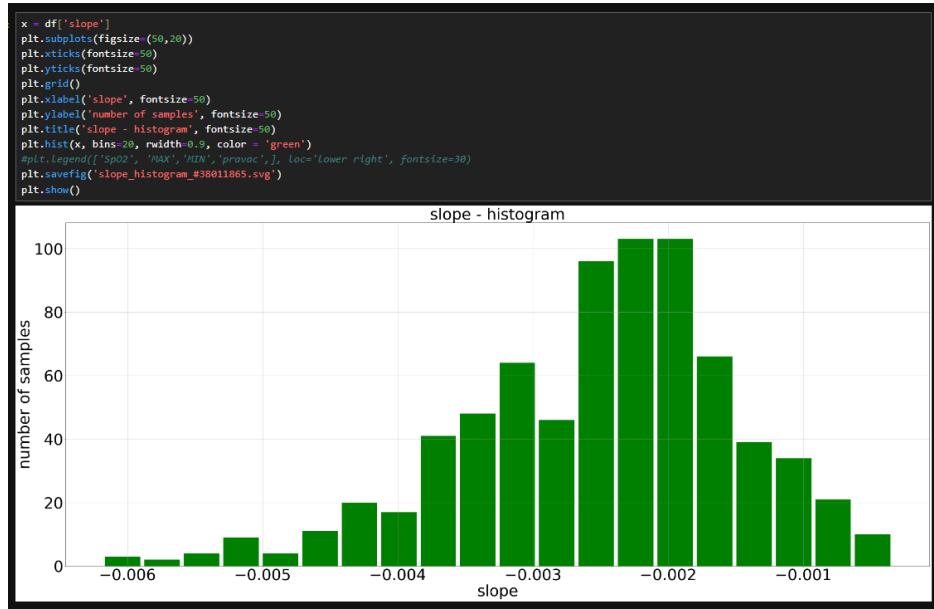
ODI (engl. *Oxygen Desaturation Index*) označava broj padova zasićenosti krvi kisikom većih od 3% i duljih od 10 sekundi. Upravo je ODI jedna od najkorisnijih i izravnijih mjera ozbiljnosti slučaja apneje s obzirom da direktno računa koliko poremećaj utječe na opskrbu kisikom. Cilj ove bilježnice je bio sam izračun ODI-ja i karakterizacija događaja klasificiranih kao OD - njihovog trajanja, intenziteta i naglosti pada zasićenosti krvi kisikom.



Slika 6.2: Izdvajanje lokalnih minimuma i maksimuma čija je razlika značajan pad u zasićenosti kisikom (>3%).



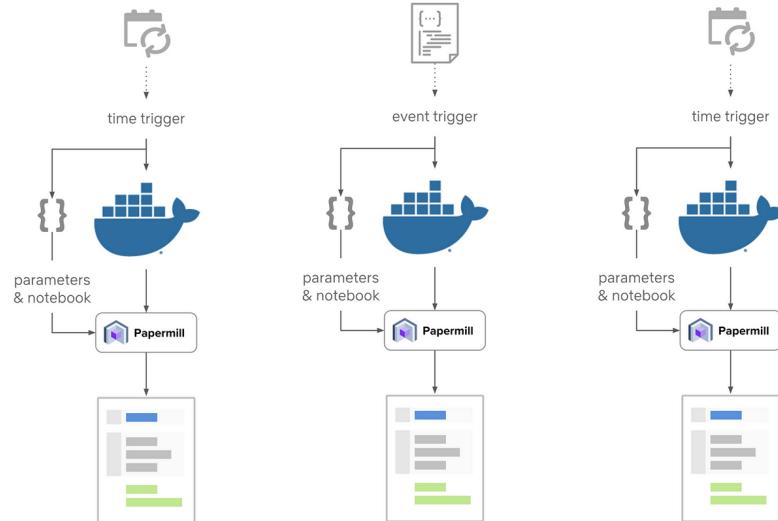
Slika 6.3: Distribucija događaja desaturacije po vremenskom trajanju.



Slika 6.4: Distribucija događaja desaturacije po nagibu krivulje zasićenja - naglosti pada razine kisika.

6.1.1. Papermill

Papermill je projekt otvorenog koda za Python, radi se o paketu koji omogućava parametrizaciju i izvršavanje parametriziranih Jupyter bilježnica. Ovakav pristup omogućuje korištenje bilježnica poput funkcija odnosno automatiziranih izvješća. Posebno je zanimljiva mogućnost korištenja Papermilla kao temelja za stvaranje **pipelinea** za obradu podataka, gdje pojedina transformira ulazne podatke u oblik potreban ulazu druge bilježnice te uz to opcionalno daje ljudima čitljivo izvješće.



Slika 6.5: Primjer korištenja Papermilla u podatkovnim pipeline-ima (slika preuzeta sa Netflix Technology Blog¹).

Pokretanje samih bilježnica koristeći parametre moguće je iz drugih bilježnica kao što je vidljivo na slici 6.6 gdje se poziva pokretanje računanja ODI-ja za sve pacijente u mapi 'Diplomski' - izlazne bilježnice se pohranjuju u mapi pacijenta.

```
import pandas as pd
import numpy as np
import csv
import os
import copy
import mne
import shutil
import papermill as pm

root_path="C:\\\\Users\\\\bruno.grbavac\\\\Desktop\\\\Diplomski"
file_name="Histogrami-ODI.ipynb"

def run_all_notebooks(dir, file):
    subdirs = [f.path for f in os.scandir(dir) if f.is_dir()]
    for subdir in subdirs:
        if(subdir!=os.path.join(dir, ".ipynb_checkpoints")):
            edf_path = os.path.join(subdir, "edf_signals.edf")
            og_notebook_path = os.path.join(dir,file)
            notebook_path = os.path.join(subdir,file)

            pm.execute_notebook(
                og_notebook_path,
                notebook_path,
                parameters=dict(edf_path=edf_path)
            )
```

Slika 6.6: Pokretanje bilježnice **Histogrami** nad signalom **edf_signals.edf** u mapama mape **Diplomski**.

Osim pokretanja kroz Jupyter bilježnice Papermill omogućava isto kroz CLI te je moguće korištenje YAML datoteka kao zbirke parametara - time se izrazito olakšava kreiranje pipelinea gdje prethodna bilježnica stvara parametre iduće.

```
$ papermill local/input.ipynb s3://bkt/output.ipynb -p alpha 0.6 -p l1_ratio 0.1
```

Slika 6.7: Pokretanje bilježnice kroz sučelje naredbenog redka.

```
$ papermill local/input.ipynb s3://bkt/output.ipynb -f parameters.yaml
```

Slika 6.8: Pokretanje bilježnice kroz sučelje naredbenog redka koristeći parametre iz YAML datoteke.

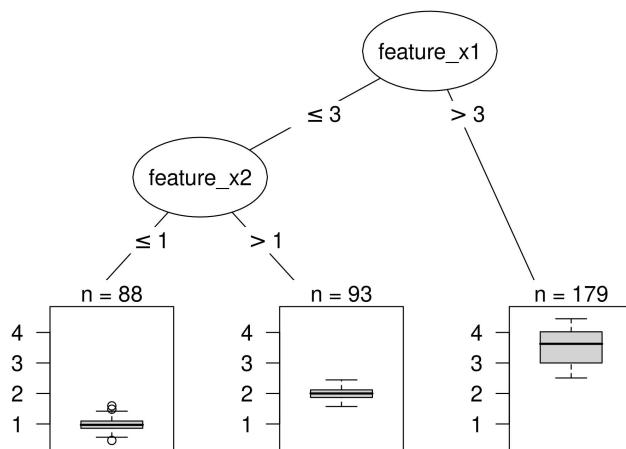
¹Netflix Technology Blog: Part 2: Scheduling Notebooks at Netflix, s Interneta: <https://netflixtechblog.com/scheduling-notebooks-348e6c14cf6>, 11.09.2023.

6.2. Random forest

Algoritam slučajnih šuma jedan je od najrasprostranjenijih algoritama strojnog učenja zbog svoje robustnosti i sposobnosti rješavanja i problema klasifikacije i problema regresije. Sam algoritam predstavili su **Leo Breiman i Adele Cutler**, a temelji se na **stablima odluke** i Breimanovom konceptu **bagging-a** kojeg je predstavio u radu 1996. godine.[17]

Sam koncept temelji se na korištenju više stabala odluke kojima dajemo različite dijelove skupa podataka, a rezultat dobivamo kao ukupnu odluku svih stabala. Životni primjer ovakvog zaključivanja bila bi situacija u kojoj 'mali Perica' odlazi do roditelja, profesora, prijatelja i drugih poznanika, svakome na malo drugčiji način ispriča svoje želje, znanja i talente kako bi mu oni preporučili buduće zanimanje te ga odabire kao zbir svih njihovih preporuka. Preporuke nisu navođene skupom podataka jer je 'Perica' svima dao po malo različitu informaciju.[18]

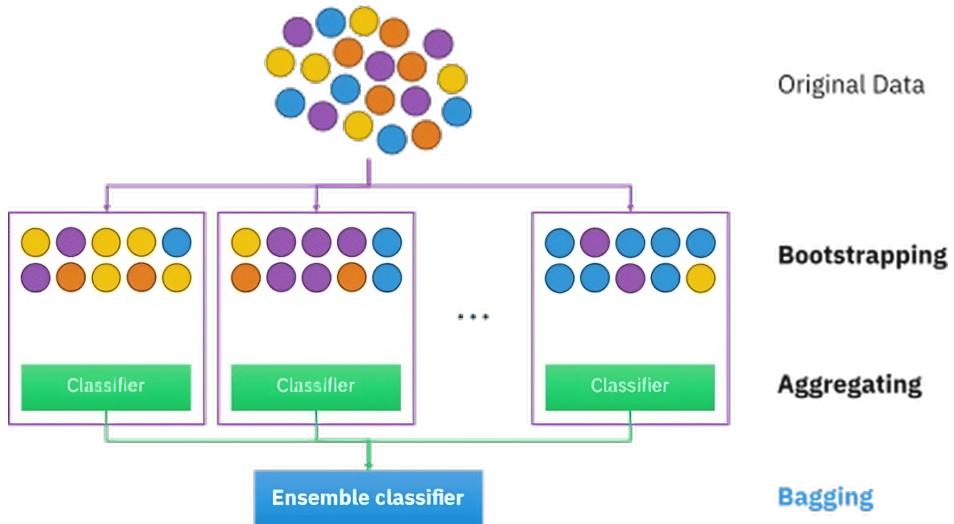
Nasumične šume sastoje se od **stabala odluke** - to su algoritmi kod kojih svaki izgrađen čvor predstavlja podjelu skupa ili podskupa podataka na osnovu vrijednosti odabrane značajke s ciljem da je tako nastala podjela što homogenija to jest da je za nju što lakše donijeti konačnu traženu odluku (računa se najčešće **Gini Index** ili **Information Gain** heuristika - za podjelu se uzima značajka sa najvećom vrijednošću). Problem stabala kao samostalne metode je izrazita sklonost **pre-traniranju** ukoliko se dopusti ne-kontrolirano grananje bez metoda poput **obrezivanja** (engl. *pruning*).[17][19]



Slika 6.9: Primjer stabla odluke (slika preuzeta sa ChristophM GitHub²).

²ChristophM GitHub: Decision Tree, s Internata: <https://christophm.github.io/interpretable-ml-book/tree.html>, 11.09.2023.

Bagging (engl. *Bootstrap aggregation*) je metoda u kojoj se pojedinom stablu šume dodjeljuje nasumično izabran skup uzoraka biran se mogućnosti ponavljanja. Nakon što su pridjeljeni takvi skupovi podataka, svako stablo se trenira zasebno te se ovisno o vrsti problema ukupna promjena bira prosječnim ili većinskim izborom.



Slika 6.10: Metoda Bootstrap agregacije. [18]

Osim *Bagging*-a skupa podataka koristi se i **bagging značajki** - metoda nasumičnog podprostora čime se svakom stablu pridružuje i nasumično odabran skup značajki sa kojima trenira - ovo dodatno smanjuje korelaciju među stablima što smanjuje ukupnu varijancu i mogućnost pre-treniranja te podiže točnost modela. Također, trećina svakog *bootstrap* skupa podataka dodijeljenog pojedinom stablu se ne koristi za treniranje već se "ostavlja sa strane" kako bi se kasnije koristila za **unakrsnu validaciju** čime se gubi potreba za testnim skupom podataka. Ukoliko je riječ o problemu klasifikacije - ukupna je odluka jednaka **izboru većine**, dok se kod regresije konačna odluka računa kao **prosječna odluka** svih stabala. [18]

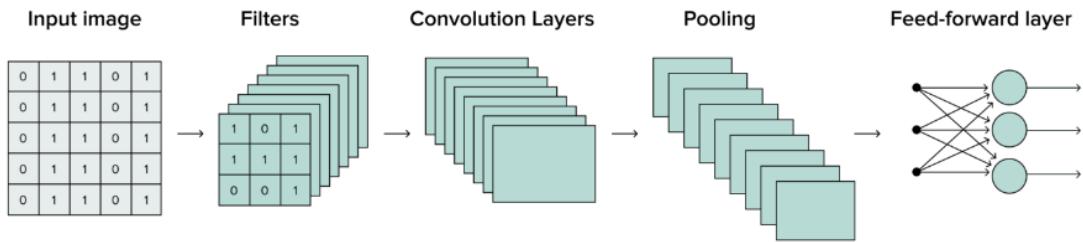
Glavna prednost šuma je njihova robustnost, stabilnosti i otpornost na pre-treniranje koje proizlaze iz donošenja odluke uprosječivanjem odluka svih stabala koja su u međusobno slaboj korelaciji. Također, s obzirom da se stabla treniraju nezavisno, izgradnju šume lako je paralelizirati te se time vrijeme treniranja uvelike približava onome potrebno za trening pojedinog stabla. Uz to ovaj model izričito uspješan na visoko-dimenzionalnim skupovima podataka, neuravnoteženim skupovima te skupovima sa nedostajućim vrijednostima. [18]

Sama primjena ovih dvaju modela prikazala se relativno neuspješnom, a razlog možemo dokučiti analizom značajki. Naime ako jednako zastupljene faze sna s normaliziranim značajkama iscrtamo, vidimo gotovo pa identične distribucije vrijednosti većine pojedinih značajki po svim fazama. Te nam značajke stoga ne doprinose ka razaznavanju faze sna tog uzorka. Ovo je lako vidljivo i iz toga što je otklanjanjem takvih značajki rezultat postignut na klasifikatorima gotovo nepromijenjen.

Iz ovoga se da zaključiti kako je za uspješnije klasificiranje ovih fazi potrebno u obzir uzeti kontekst uzorka to jest susjedne uzorke i njihove vrijednosti.

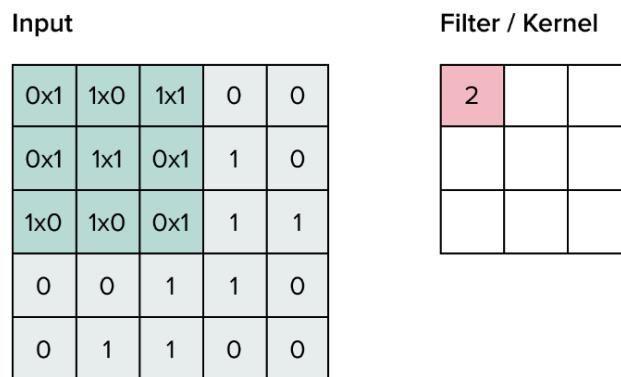
6.3. Konvolucijske neuronske mreže

Konvolucijske mreže (engl. skraćeno **CNN**) su modeli često korišteni za potrebe računalnog vida, ali i obrade prirodnog jezika te klasifikacije. Glavna je karakteristika ovih modela prepoznavanje prostorno ovisnih uzoraka u podacima. Ovakva mreža najčešće se sastoji od 5 slojeva: **ulaznog** sloja - podaci koje obrađujemo, **konvolucijskog** sloja, **pooling** sloja, **potpuno povezanog** sloja te **izlaznog** sloja u kojem je odluka o rješenju problema. [20]



Slika 6.11: Slojevi konvolucijske neuronske mreže (slika preuzeta sa Serokell³).

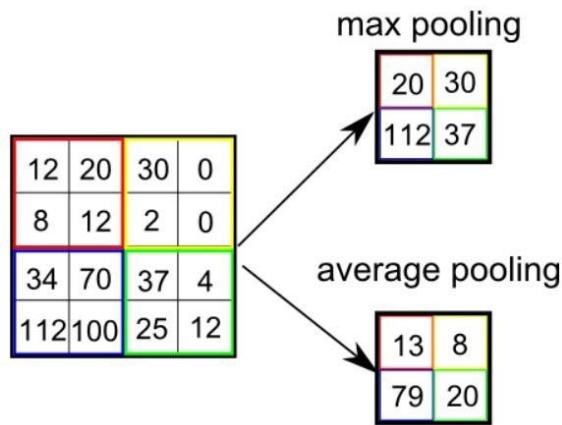
Sama **konvolucija** po kojoj su mreže nazvane je operacija množenja članova dvaju matrica istih dimenzija te potom njihovo zbrajanje. Ova operacija je temelj konvolucijskog sloja gdje se konvolucijom dijelova ulaznih podataka i **filtera** - matrice postavljene za izvlačenje informacija o pojedinoj značajki, derivira informacija o spacialnoj korelaciji ulaznih podataka. Time se dobiva **mapa značajki** koja sadrži apstrakciju informacije o međusobnim prostornim odnosima ulaznih podataka. [20]



Slika 6.12: Primjena konvolucijskog filtra na ulaz - operacija konvolucije (slika preuzeta sa Serokell³).

³Serokell: Convolutional Neural Networks for Beginners, s Interneta: <https://serokell.io/blog/introduction-to-convolutional-neural-networks>, 11.09.2023.

Pooling sloj zadužen je za redukciju dimenzija mape značajki čime se ubrzava rad mreže, s obzirom da smanjuje broj parametara koje je potrebno procesirati. Pooling može biti **MAX** gdje se grupa članova mape značajki zamjenjuje svojim najvećim članom, ili **AVG** gdje se zamjenjuje prosječnom vrijednošću članova. Rezultat rada konvolucijskog i *pooling* sloja je apstrakcija značajki ulaznih podataka matematičkim pravilima same mreže. [20]

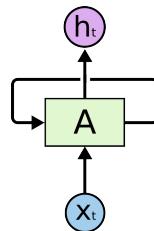


Slika 6.13: Operacija poolinga po maksimumu i prosjeku. [20]

Ovakve mreže se s obzirom na mogućnost prepoznavanja i izdvajanja prostornih uzoraka (oblika) uglavnom koriste u obradi slika - sami filter se pomiće u 2 dimenzije po slici, kako bi se konvolucijom pokrili svi djelovi slike. Ovakve mreže se zbog takvog pomicanja filtera nazivaju **2D konvolucijske mreže**. Osim takvih postoje, te su nama za ovaj problem vrlo zanimljive, **1D konvolucijske mreže** koje su primjenu našle u izdvajaju apstrakciji uzoraka (izvlačenje trendovskih značajki signal) jedno-dimenzionalnih podataka poput audio zapisa, videa, tekstova (obradi prirodnog jezika) i općenito obradi **vremenskih nizova** (kao što je korišteni signal saturacije krvi kisikom). [21]

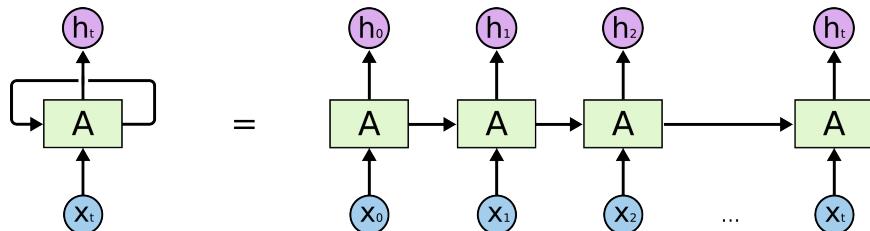
6.4. LSTM (engl. *Long Short-Term Memory*) mreže

Iako su temeljene upravo na ljudskom živčanom sustavu zaduženom za razmišljanje, obične neuronske mreže (engl. Multi Perceptron) imaju manu u činjenici što previđaju jedan dio prirode ljudskog razmišljanja i učenja, a to je da **ljudi uče uz kontekst prethodno usvojenog** znanja. Ovaj previd ispravljaju **rekurentne neuronske mreže** koje imaju moć očuvanja naučene informacije. [22]



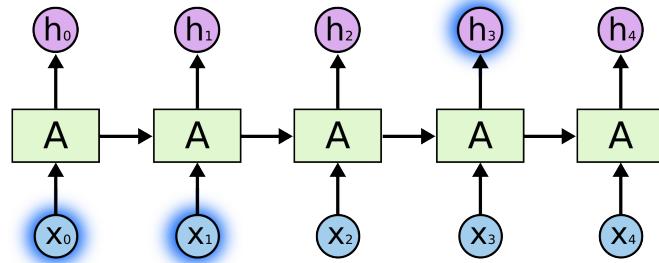
Slika 6.14: Ilustracija rekurentne neuronske mreže. [22]

Ovakve mreže najlakše je zamisliti kako informaciju održavaju unutar petlje, koju možemo "*razviti u vremenu*" i time dobivamo "**čvorove mreže**" koji naučene informacije održavaju prosljeđujući ih vremenskom **sljedbeniku**. Ovakav pogled na RNN mreže daje naslutiti da su po samoj prirodi prigodne za korištenje na podatke u obliku nizova, te u zadacima omogućavaju očuvanje konteksta - uče povezati te informacije sa onima dalje u mreži.

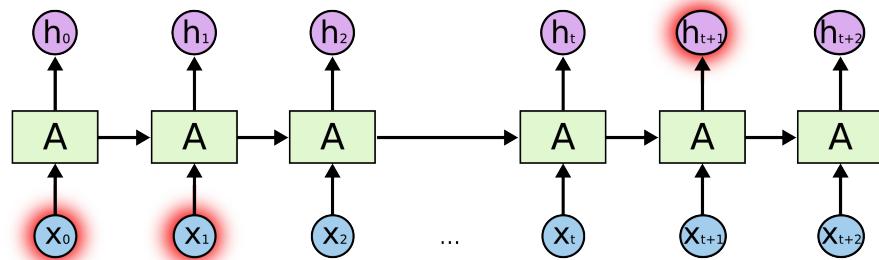


Slika 6.15: Rekurentna neuronska mreža kao lanac vremenski raspoređenih ćelija. [22]

Iako iznimno korisne, obične RNN imaju problem sa učenjem šireg konteksta, izrazito su efikasne u povezivanju informacija naučenih u relativno bliskim "*čvorovima*" mreže, no nemaju mogućnost učenja "**dugotrajnih zavisnosti**", to su u svojim radovima pokazali nijemac **Hochreiter** i talijan **Bengio** početkom devedesetih godina prošlog stoljeća. [22]

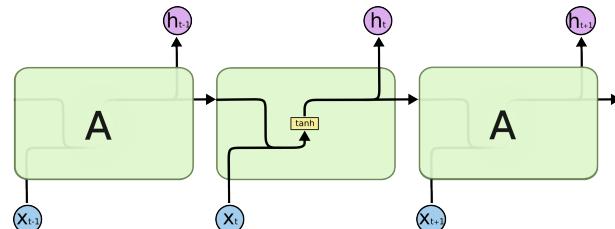


Slika 6.16: RNN mreže uspješno uče povezanost među bliskim podacima (uzak kontekst). [22]

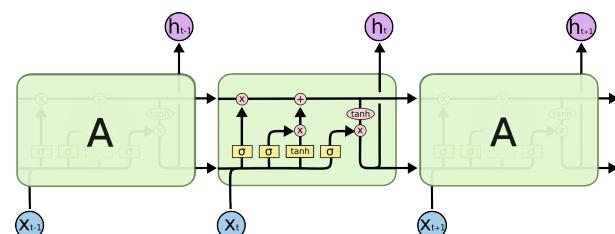


Slika 6.17: RNN mreže ne uspijevaju naučiti zavisnosti između jako razmaknutih podataka. [22]

Upravo će 1997. **Hochreiter** i **Schmidhuber** predstaviti mrežu sposobnu za učenje takvih zavisnosti. Za razliku od običnih RNN, **LSTM** mreže temelje se na 4 sloja neuronske mreže, posložena kako bi pamćenje dugih konteksta bilo prirodno ponašanje mreže. [22]



Slika 6.18: Građa RNN mreže. [22]



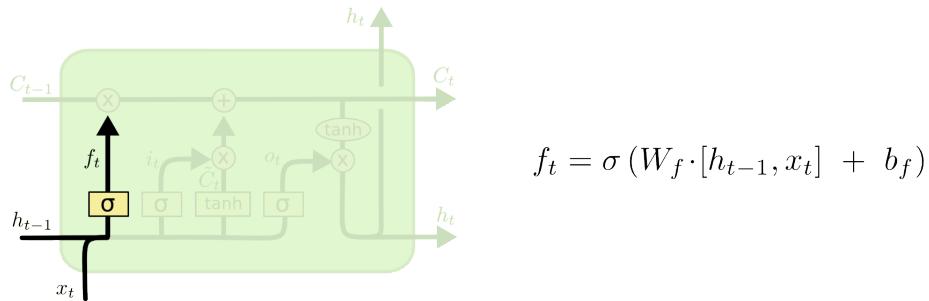
Slika 6.19: Građa LSTM mreže. [22]

Rad ćelije mreže u određenom vremenu se temelji na 3 signala:

- C_{t-1} **stanju ćelije** - trenutno dugoročno pamćenje mreže
- h_{t-1} **skrivenom stanju** - izlazna vrijednost ćelije prethodne u vremenu
- x_t **ulaznih podataka** u tom trenutku

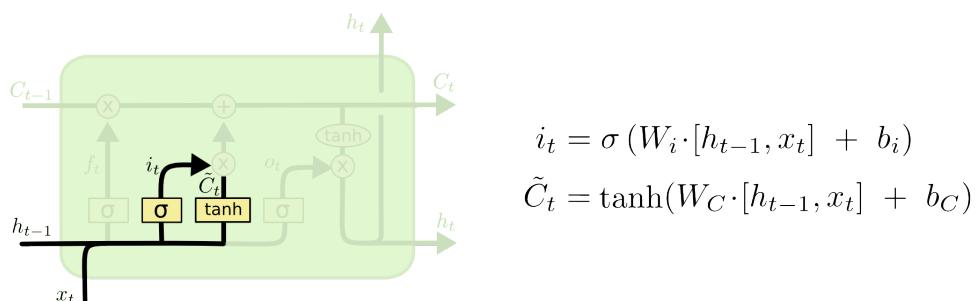
Samo stanje ćelije odnosno pamćenje mreže, možemo zamisliti kao sabirnicu koja povezuje sve ćelije mreže, pritom svaka vremenski sljedeća ćelija može **dodavati i uklanjati informacije** sa sabirnice kroz **strukture vrata** i tako modificirati pamćenje mreže.

Prvi sloj ćelije sadrži **vrata za zaboravljanje** (engl. *forget gate layer*) - na osnovu skrivenog stanja i ulaza u tom trenutku **sigmoid** vraća vrijednost od 0 do 1 koja govori **koliki dio** informacije stanja ćelije se **zadržava**.

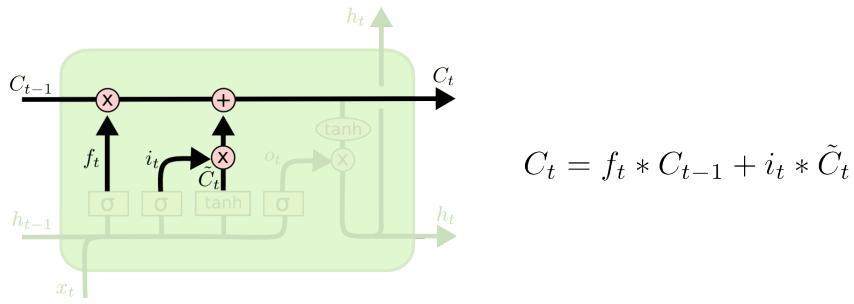


Slika 6.20: Sloj sa vratima za zaboravljanje. [22]

Drugi sloj sadrži **ulazna vrata** (engl. *input gate layer*) ćelije i njegova je zadaća proizvesti nove informacije koje će se pohraniti u stanje ćelije. Ova vrata sastoje se od 2 dijela, **sigmoida** koji odlučuje koje vrijednosti stanja ćelije mijenjamo i **tangensa hiperboličkog** koji daje **vektor kandidata** koji skaliran produktom sigmoida govori za koliko odabrane vrijednosti mijenjamo.

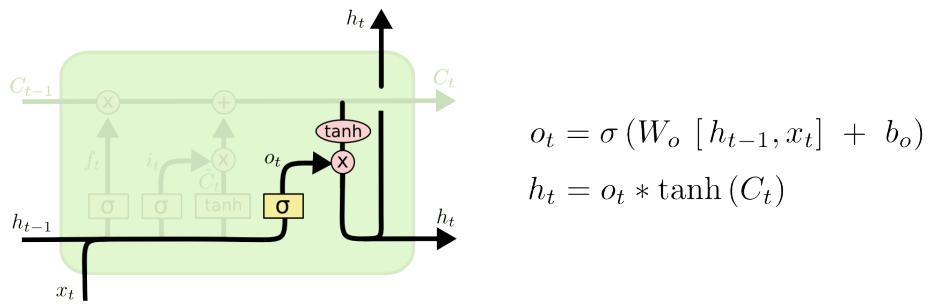


Slika 6.21: Sloj sa vratima za ulaz. [22]



Slika 6.22: Primjena promjena sloja za zaboravljanje i sloja za ulaz na stanje ćelije. [22]

Posljednji sloj sadrži **izlazna vrata** (engl. *output gate layer*) i odlučuje o informaciji koja je izlaz te ćelije odnosno skriveno stanje ćelije sljedbenika. Izlaz ćelije zapravo je odabrani dio, kroz prethodna dva sloja novo-nastalog, stanja ćelije. Kao i u prethodna dva sloja, za odabir dijelova stanja koristi se sigmoid dok se stanje ćelije prethodno skalira kroz tangens hiperbolički.



Slika 6.23: Sloj sa izlaznim vratima. [22]

6.5. Klasifikacija faze sna

6.5.1. Analiza značajki

Praćenje ispravne izmjene ne-REM i REM fazi sna jedan je od glavnih uvida u zdravlje spavanja. Kako bi se olakšali zahtjevi na resurse treniranja prvo je napravljena jako jednostavna analiza značajki. Uravnotežene su klase skupa podataka te su same vrijednosti normalizirane i kvadrirane kako bi se lakše uočila različite distribucije.

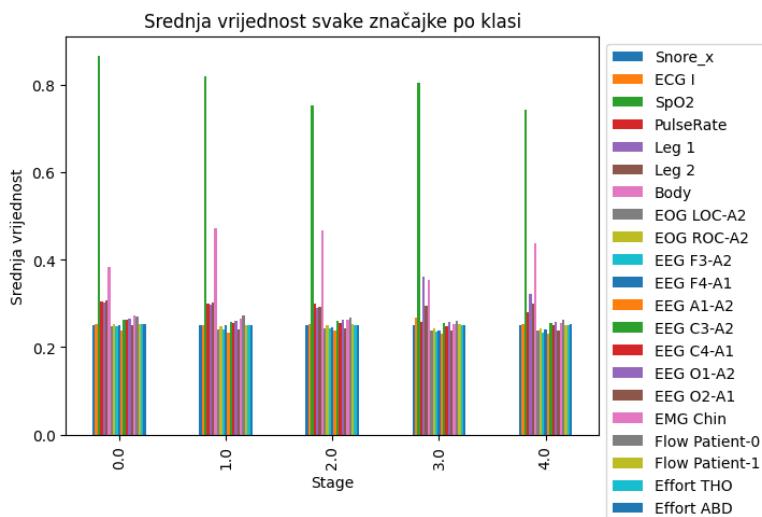
```
[ ] edf = pd.DataFrame()
for i in range (1,23):
    path = edf_path + str(i) + '.feather'
    edf_temp = feather.read_table(path).to_pandas()
    edf = pd.concat([edf, edf_temp.iloc[:,100:,:]])
    
    [ ] plot_edf = edf[['Snore_X', 'ECG I', 'SpO2', 'PulseRate', 'Leg 1', 'Leg 2', 'Body', 'Stage','EOG LOC-A2',
    'EOG ROC-A2', 'EEG F3-A2', 'EEG F4-A1', 'EEG A1-A2', 'EEG C3-A2', 'EEG C4-A1', 'EEG O1-A2',
    'EEG O2-A1', 'EMG Chin','Flow Patient-0', 'Flow Patient-1', 'Effort THO', 'Effort ABD']]

    min_stage_len = plot_edf['Stage'].value_counts().min()
    plot_edf = plot_edf.groupby('Stage').apply(lambda x: x.sample(min_stage_len)).reset_index(drop=True)

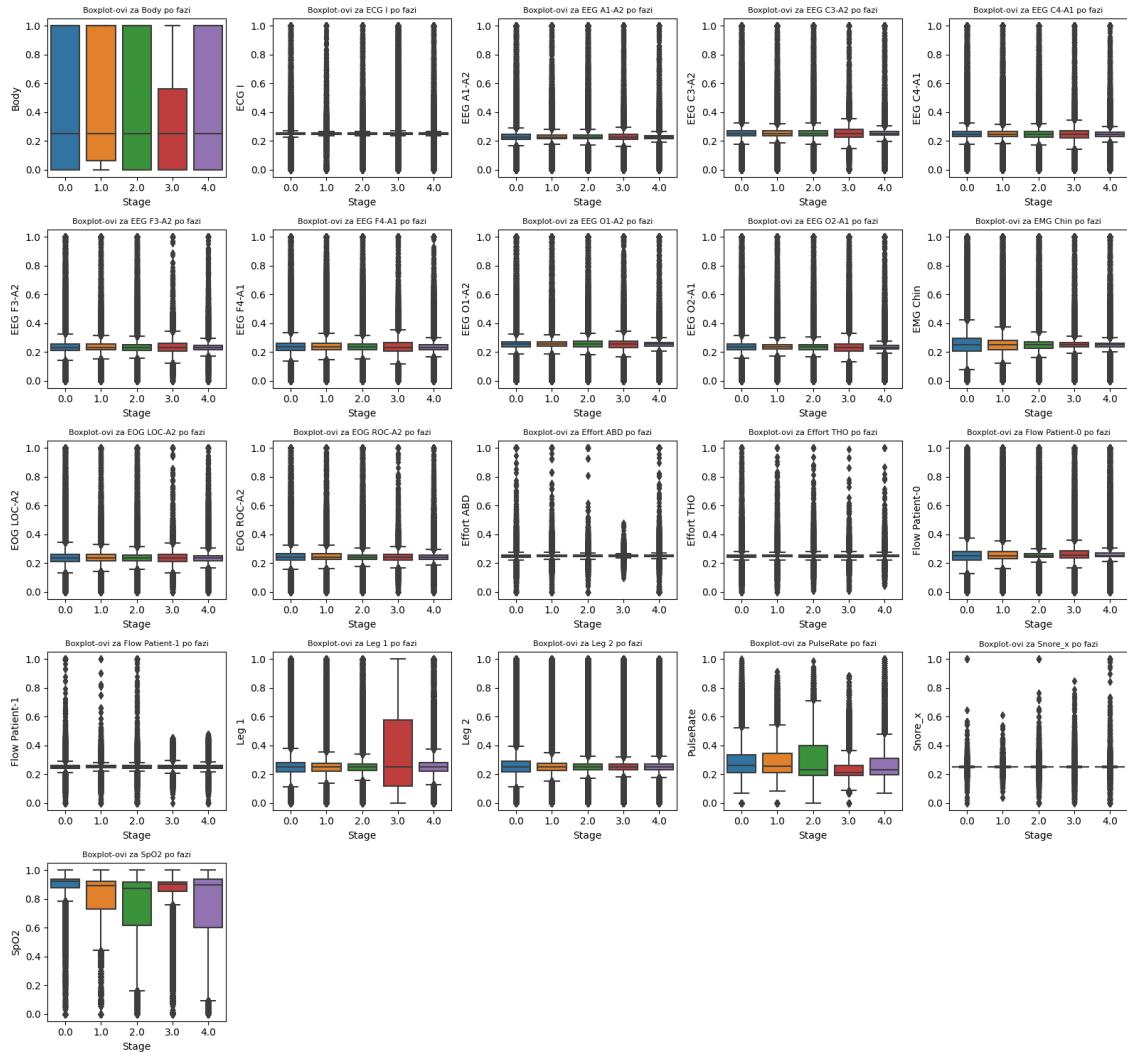
    plot_edf = (plot_edf - plot_edf.min()) / (plot_edf.max() - plot_edf.min())
    plot_edf = plot_edf.pow(2)

    plot_edf['Stage'] = plot_edf['Stage'].pow(0.5)*4
```

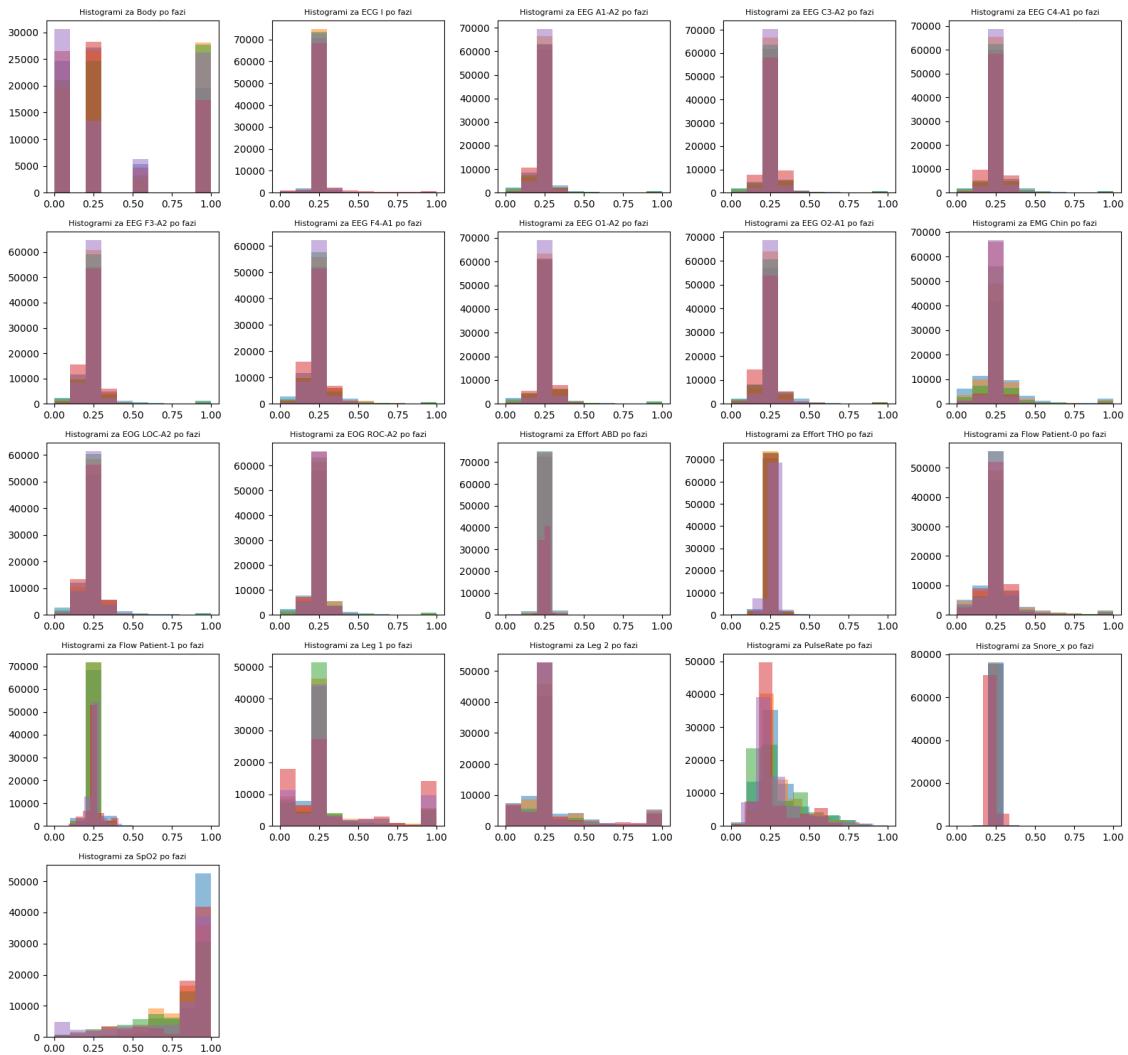
Slika 6.24: Učitavanje skupa podataka, izlučivanje stupaca polisomnografije, uravnotežavanje po klasama te normalizacija vrijednosti.



Slika 6.25: Prosječne vrijednosti značajki polisomnografije po fazama sna.



Slika 6.26: Box dijagrami vrijednosti značajki po fazama sna.



Slika 6.27: Histogrami distribucija vrijednosti značajki po fazama sna.

Iz priloženih grafova da se uočiti da se **distribucije nekoliko značajki razlikuju ovisno o klasi**, dok kod drugih one ostaju nepromijenjene. Za razvoj klasifikatora upravo su nam bitni podaci iz čijih se vrijednosti - koje se mijenjaju ovisno o klasi da donijeti zakonitost za donošenje odluke. Uočene značajke su: 'Flow Patient-1', 'Effort THO', 'Effort ABD', 'Snore_x', 'SpO2', 'PulseRate', 'Body', 'Leg 1'. Prvotna pretpostavka bila je da je moguće, na osnovu tih

nekoliko značajki ovog skupa, zaključiti o trenutnoj fazi na osnovu samo toga jednog uzorka bez konteksta okolnih uzoraka. Stoga su postavljena 3 modela - 2 modela, prethodno opisani Random Forest klasifikator i klasifikator temeljen na pojačavanju gradijenta zvan **xGBoost**.

Prvi postavljen model je *Random Forest* iz Python biblioteke **scikit-learn** sa 100 stabala i pokazuje nezavidne rezultate vidljive na slici 6.28, iz čega se već dalo posumnjati na to da će u procjeni faze sna, biti potrebni okolni uzorci - tj. **kontekst zaključivanja**.

```
[ ] rfclf = RandomForestClassifier(n_estimators=100, n_jobs=-1)
rfclf.fit(x_train, y_train)

[ ] y_pred = rfclf.predict(x_test)

print("Točnost: ", accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

Točnost: 0.7352572963668136
[[11343 1441 1161 595 800]
 [ 1445 11478 1327 509 637]
 [ 1575 1894 8624 1365 1888]
 [ 446 344 606 13586 447]
 [ 900 723 1578 620 11350]]
      precision    recall   f1-score   support
      0       0.72     0.74     0.73    15340
      1       0.72     0.75     0.73    15396
      2       0.65     0.56     0.60    15346
      3       0.81     0.88     0.85    15429
      4       0.75     0.75     0.75    15171

      accuracy         0.74    76682
      macro avg       0.73     0.74     0.73    76682
      weighted avg    0.73     0.74     0.73    76682
```

Slika 6.28: Random Forest model nad 8 značajki, bez konteksta susjednih uzoraka.

Model temeljen na **pojačavanju gradijenta XGBoost**, sa slike 6.29 također nije dao neke zavidne rezultate, neovisno o pokušajima podešavanja parametara.

```
[ ] xgbClf = XGBClassifier(
    n_estimators=100,
    n_jobs=-1,
    max_depth=12,
    learning_rate=0.5
)
xgbClf.fit(X_train, y_train)

[ ] y_pred = xgbClf.predict(X_test)

print("Točnost: ", accuracy_score(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))

Točnost: 0.7025507941889882
[[18914 1672 1380 588 866]
 [ 1624 10723 1759 512 778]
 [ 1591 2079 8211 1396 2069]
 [ 492 421 789 13179 548]
 [ 945 847 1880 653 10846]]
      precision    recall   f1-score   support
      0       0.70      0.71      0.71      15340
      1       0.68      0.70      0.69      15396
      2       0.59      0.54      0.56      15346
      3       0.81      0.85      0.83      15429
      4       0.72      0.71      0.72      15171

      accuracy         0.70      76682
      macro avg       0.70      0.70      0.70      76682
      weighted avg    0.70      0.70      0.70      76682
```

Slika 6.29: XGBoost model nad 8 značajki, bez konteksta susjednih uzoraka.

Pokušaj je zatim bio napisati model **konvolucijske neuronske mreže** za klasifikaciju, koristeći **Tensorflow Keras** biblioteku, te je cilj bio - ovisno o rezultatima zaključiti je su li **klasifikacija faze sna uzorka kontekstualno ovisna** o susjednim odnosno prethodnim uzorcima. Kao što je vidljivo na slici 6.30 mreža je dala poprilično točne klasifikacije, te se iz toga može zaključiti da je prethodnim modelima bilo potrebno pružiti kontekst - okvir susjednih vrijednosti uz pomoć kojih bi donijeli odluku.

```
[ ] model = Sequential()

model.add(Conv1D(filters=64, kernel_size=4, activation='relu', input_shape=input_shape))
model.add(MaxPooling1D(pool_size=2))

model.add(Conv1D(filters=32, kernel_size=4, activation='relu'))
model.add(MaxPooling1D(pool_size=2))

model.add(Flatten())

model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

model.summary()

[ ] test_loss, test_acc = model.evaluate(X_test_conv, y_test_conv)

print(f"Točnost: {test_acc}")

2397/2397 [=====] - 395 16ms/step - loss: 0.1252 - accuracy: 0.9531
Točnost: 0.9531001448631287
```

Slika 6.30: Predviđanje faze na temelju CNN mreže.

Dodajući kontekst podacima korištenim na prva dva klasifikatora - **standardna devijacija i prosjek** prethodnih 20 uzoraka, postignute su izvrsne točnosti klasificiranja vidljive na slici 6.31

```
[ ] def context_window(x, window_size=20):
    X_context = []
    for i in range(len(x)):
        start = max(0, i - window_size)
        end = min(len(x), i + window_size + 1)
        context = x.iloc[start:end]

        context_mean = context.mean().values
        context_std = context.std().values

        features = np.concatenate([x.iloc[i].values, context_mean, context_std])
        X_context.append(features)

    return np.array(X_context)

X_context = context_window(X_scaled)
```

```
[ ] y_pred = rfCtxClf.predict(X_test_ctx)
print("Točnost: ", accuracy_score(y_test_ctx, y_pred))
print(confusion_matrix(y_test_ctx, y_pred))
print(classification_report(y_test_ctx, y_pred))

[Parallel(n_jobs=8)]: Using backend ThreadingBackend with 8 concurrent workers.
[Parallel(n_jobs=8)]: Done 25 tasks      | elapsed:  0.1s
Točnost:  0.9882240943115724
[[15342   51    1    0    3]
 [ 111 15177   189    0    9]
 [   2   215 14934    0   132]
 [   0     0    3 15231    0]
 [   2     2   183    0 15095]]
      precision    recall   f1-score   support
          0       0.99     1.00     0.99    15397
          1       0.98     0.98     0.98    15486
          2       0.98     0.98     0.98    15283
          3       1.00     1.00     1.00    15234
          4       0.99     0.99     0.99    15282

      accuracy                           0.99
     macro avg       0.99     0.99     0.99    76682
  weighted avg       0.99     0.99     0.99    76682

[Parallel(n_jobs=8)]: Done 100 out of 100 | elapsed:  0.4s finished
```

```
[ ] y_pred_xgbctx = xgbClfCtx.predict(X_test_ctx)

print("Točnost: ", accuracy_score(y_test_ctx, y_pred_xgbctx))
print(confusion_matrix(y_test_ctx, y_pred_xgbctx))
print(classification_report(y_test_ctx, y_pred_xgbctx))

Točnost:  0.9964789650765499
[[15368   26    2    0    1]
 [ 38 15407   39    0    2]
 [   2   50 15192    0   39]
 [   0     0    0 15234    0]
 [   2     2   67    0 15211]]
      precision    recall   f1-score   support
          0       1.00     1.00     1.00    15397
          1       0.99     0.99     0.99    15486
          2       0.99     0.99     0.99    15283
          3       1.00     1.00     1.00    15234
          4       1.00     1.00     1.00    15282

      accuracy                           1.00
     macro avg       1.00     1.00     1.00    76682
  weighted avg       1.00     1.00     1.00    76682
```

Slika 6.31: Dodavanje kontekstualnog prozora i rezultati modela u kojima je primjenjen..

S obzirom da su u ovim primjerima korišteni i svakodnevno nedostupni senzori, poput torakalnih i abdomenskih traka i senzora za tok zraka, pokušano je postići prihvatljive rezultate koristeći sa što manje takvih značajki. Pokušaj se pokazao poprilično uspješan kao što se vidi iz slika 6.33 koristeći mjerjenja sa mikrofona za hrkanje, senzora saturacije krvi kiskom i senzora za otkucaje srca, koji su svi danas dostupni na rješenjima poput pametnih satova, pa bi se ovakva klasifikacija mogla izvršavati za svakodnevnu kontrolu kvalitete sna.

```

[ ] psg_columns=['Snore_X', 'PulseRate','SpO2']

# psg_columns=['Snore_X', 'ECG_I', 'SpO2', 'PulseRate', 'Leg_1', 'Leg_2', 'Body','EOG_LOC-A2', 'EOG_ROC-A2', 'EEG_F3-A2', 'EEG_F4-A1', 'EEG

[ ] y_pred = rfCtxClf.predict(X_test_ctx)
print("Točnost: ", accuracy_score(y_test_ctx, y_pred))
print(confusion_matrix(y_test_ctx, y_pred))
print(classification_report(y_test_ctx, y_pred))

Točnost:  0.9669022717195691
[[15127  241   6    0   23]
 [ 275 14722  470    0   19]
 [   5  433 14388    0  457]
 [   0    1   4 15226    3]
 [  32   24   545    0 14681]]
      precision    recall   f1-score   support
      0       0.98     0.98     0.98    15397
      1       0.95     0.95     0.95    15486
      2       0.93     0.94     0.94    15283
      3       1.00     1.00     1.00    15234
      4       0.97     0.96     0.96    15282

      accuracy                           0.97    76682
     macro avg       0.97     0.97     0.97    76682
  weighted avg       0.97     0.97     0.97    76682

[ ] y_pred_xgbctx = xgbClfCtx.predict(X_test_ctx)

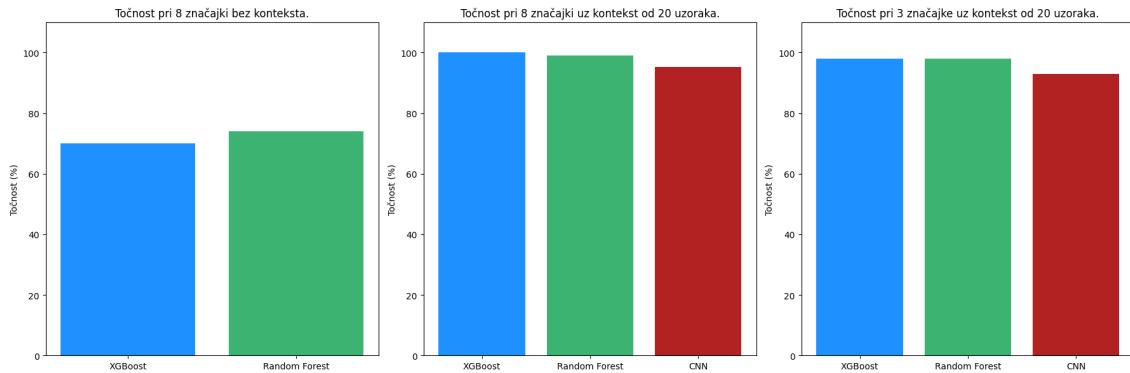
print("Točnost: ", accuracy_score(y_test_ctx, y_pred_xgbctx))
print(confusion_matrix(y_test_ctx, y_pred_xgbctx))
print(classification_report(y_test_ctx, y_pred_xgbctx))

Točnost:  0.9782347878250437
[[15208  170   6    0   13]
 [ 185 14988  298    1  14]
 [   1  287 14697    0 298]
 [   0    1   4 15229    0]
 [  15   18   358    0 14891]]
      precision    recall   f1-score   support
      0       0.99     0.99     0.99    15397
      1       0.97     0.97     0.97    15486
      2       0.96     0.96     0.96    15283
      3       1.00     1.00     1.00    15234
      4       0.98     0.97     0.98    15282

      accuracy                           0.98    76682
     macro avg       0.98     0.98     0.98    76682
  weighted avg       0.98     0.98     0.98    76682

```

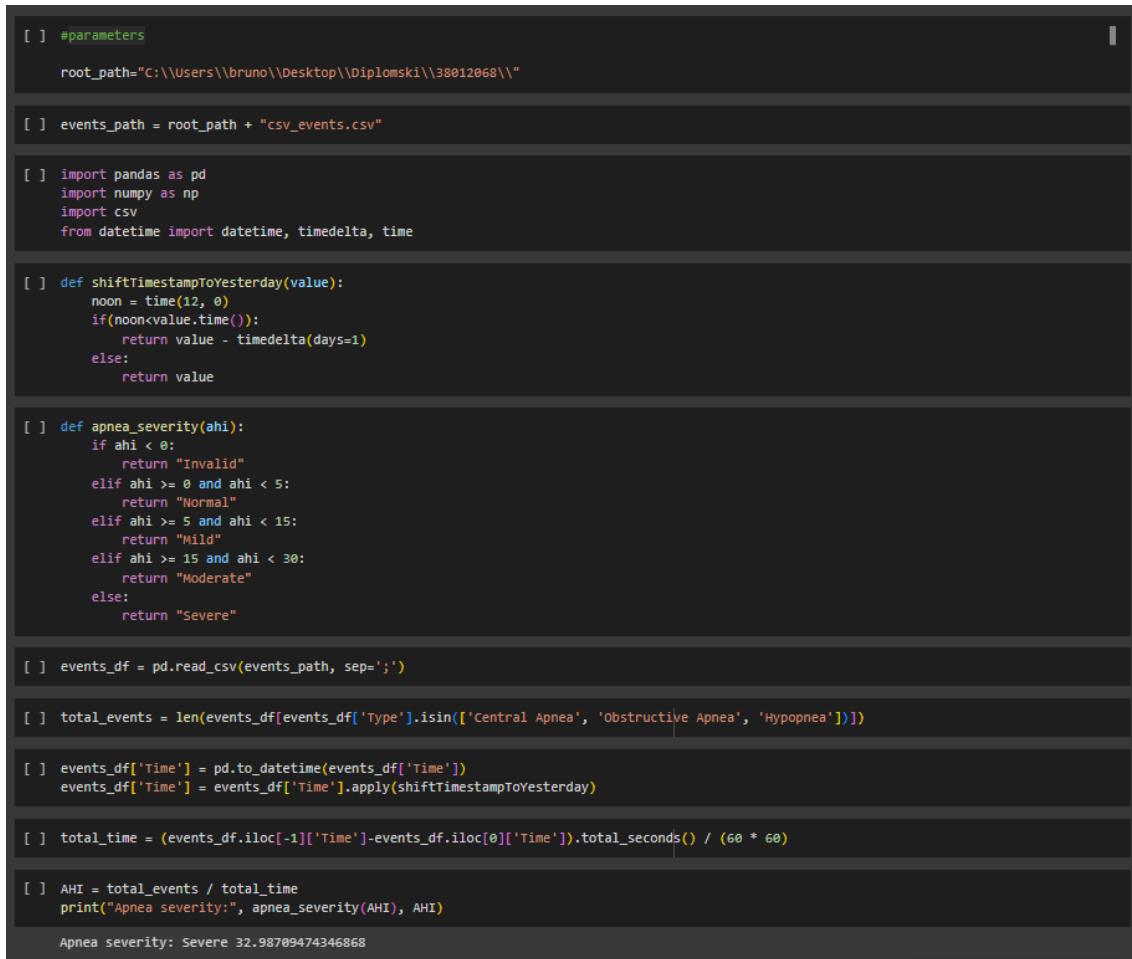
Slika 6.32: Korištenje samo 3 značajke za klasifikaciju faze sna.



Slika 6.33: Točnosti implementiranih modela za klasifikaciju faze sna.

6.6. Predviđanje AHI indeksa

Kao što je prethodno spomenuto, AHI indeks se koristi za ocjenu ozbiljnosti oboljenja apneje. AHI za pojedini zapis u slučaju korištenog skupa podataka radi se koristeći *csv_events* datoteke, brojeći događaje apneje i hipopneje te dijeleći ih sa trajanjem samog zapisa. Kalkulator AHI-ja prikazan je na slici 6.34.



```
[ ] #parameters
root_path="C:\\\\users\\\\bruno\\\\Desktop\\\\Diplomski\\\\38012068\\\\"

[ ] events_path = root_path + "csv_events.csv"

[ ] import pandas as pd
import numpy as np
import csv
from datetime import datetime, timedelta, time

[ ] def shiftTimestampToYesterday(value):
    noon = time(12, 0)
    if(noon < value.time()):
        return value - timedelta(days=1)
    else:
        return value

[ ] def apnea_severity(ahi):
    if ahi < 0:
        return "Invalid"
    elif ahi >= 0 and ahi < 5:
        return "Normal"
    elif ahi >= 5 and ahi < 15:
        return "Mild"
    elif ahi >= 15 and ahi < 30:
        return "Moderate"
    else:
        return "Severe"

[ ] events_df = pd.read_csv(events_path, sep=';')

[ ] total_events = len(events_df[events_df['Type'].isin(['Central Apnea', 'Obstructive Apnea', 'Hypopnea'])])

[ ] events_df['Time'] = pd.to_datetime(events_df['Time'])
events_df['Time'] = events_df['Time'].apply(shiftTimestampToYesterday)

[ ] total_time = (events_df.iloc[-1]['Time']-events_df.iloc[0]['Time']).total_seconds() / (60 * 60)

[ ] AHI = total_events / total_time
print("Apnea severity:", apnea_severity(AHI), AHI)

Apnea severity: Severe 32.98709474346868
```

Slika 6.34: Jupyter bilježnica za izračun AHI indeksa pojedinog pacijenta.

Koristeći **izračunate indekse** i zapise pacijenata o **zasićenosti krvi kisikom** pokušano je izračunati indeks bez korištenja zapisa o samim događajima čime izbjegavamo kompleksne proračune temeljene na razini zasićenosti krvi kisikom. Koristeći konvolucijsku neuronsku mrežu postignute su za ovaj slučaj prihvatljivi rezultati. AHI indeks je reda veličine 10-60, dok je model dao **prosječnu kvadratnu pogrešku** od 10.78 (slika 6.35). Takva pogreška naravno nije zanemariva, no pokušaji smanjivanja iste podešavanjem parametara mreže te dodavanjem novih slojeva kako bi se moglo *naučiti više detalja*, nisu izvojevali priželjkivanim rezultatima. Ovakvi rezultati i nemogućnost poboljšanja upućuju na mali broj zapisa u skupu podataka - model je treniran na samo 18 primjera.

```

[ ] edf_path="/content/drive/MyDrive/temp2/signal_sync_EDF"
[ ] edf = pd.DataFrame()

for i in range (2,24):
    path = edf_path + str(i) + '.feather'
    edf_temp = feather.read_table(path).to_pandas()
    edf_temp['Patient'] = i
    edf = pd.concat([edf, edf_temp.iloc[:100,:]])

[ ] y = np.array([ 49.2192559545256, 79.17007141080755, 31.33331048898331, 36.30885122410546, 70.04614242844697,
51.50981226076919, 24.066003807911994, 24.066003807911994, 44.145334434351774, 23.339089339271105,
56.37456501107244, 56.37456501107244, 39.90900644003716, 29.995583689091713, 33.02068965517241,
51.50981226076919, 22.309859154929576, 43.288490284005974, 61.88520676153185, 18.013110808499857,
40.34420652391435, 32.98709474346868
])

[ ] max_length = edf.groupby('Patient')['SpO2'].count().max()
X = [np.pad(edf[edf['Patient'] == patient]['SpO2'].values, (0, max_length - len(edf[edf['Patient'] == patient]))) for patient
X = np.array(X)

[ ] X = np.expand_dims(X, axis=2)
y = np.array(y)

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] keras.utils.set_random_seed(42)

[ ] model = Sequential([
    Conv1D(32, 3, activation='relu', input_shape=(max_length, 1)),
    BatchNormalization(),
    Conv1D(64, 3, activation='relu'),
    BatchNormalization(),
    Dropout(0.2),
    Flatten(),
    Dense(64, activation='relu'),
    BatchNormalization(),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dense(1)
])

[ ] model.compile(optimizer='adam', loss='mse', metrics=['mae'])

[ ] history = model.fit(X_train, y_train, epochs=100, batch_size=6, validation_split=0.2)

[ ] loss, mae = model.evaluate(X_test, y_test)
print(f"Test Loss: {loss}")
print(f"Test MAE: {mae}")

1/1 [=====] - 0s 32ms/step - loss: 204.3757 - mae: 10.7826
Test Loss: 204.3756561279297
Test MAE: 10.78261661529541

[ ] y_pred = model.predict(X_test)
y_pred

1/1 [=====] - 0s 23ms/step
array([[57.748943],
       [31.567326],
       [21.183712],
       [58.693832],
       [51.883602]], dtype=float32)

[ ] y_test
array([49.21925595, 29.99558369, 44.14533443, 79.17007141, 51.50981226])

```

Slika 6.35: Predviđanje AHI indeksa koristeći CNN.

6.7. Predviđanje razine SpO_2

Za predviđanje razine zasićenja krvi korištena je LSTM mreža sa prozorima koji uključuju uzorke posljednjih 20 sekundi uzoraka. Predviđanje je zamišljeno na način da se **predviđa idućih 20 sekundi** na osnovu prijašnjih stvarnih mjerena (na slici 6.37 vidi se predviđanje posljednje četvrtine sesije). Ovakvim predviđanjima uspješno bi se izbjeglo gušenje pacijenta njegovim buđenjem ili pravovremenim podešavanjem CPAP terapije, čime se sprječavaju opasne posljedice poput oštećenja organa uslijed nedostatka kisika. Uz više uzoraka i resursa, za vjerovati je da bi se ovakva mreža mogla dovesti do mogućnosti da predviđa i razine nekoliko minuta unaprijed, čime bi dodatno poboljšali prevenciju epizoda gušenja u snu. Sustav na kojemu je treniran model ima 31GB radne memorije te grafičku karticu sa 15GB, što ograničava treniranje s većim prozorima predviđanja i konteksta. Na slici 6.36 je primjer predviđanja sa **prosječnom kvadratnom greškom od 0.3166004717350006**.

```
[ ] train_patients, test_patients = train_test_split(patients, test_size=3, random_state=42)

[ ] def create_dataset(patient_data, look_back=1):
    dataX = []
    for i in range(len(patient_data)-look_back-1):
        dataX.append(patient_data[i:i+look_back])
        dataX.append(patient_data[i+look_back])
    return np.array(dataX), np.array(dataY)

[ ] look_back = 100
scaler = MinMaxScaler(feature_range=(0, 1))

[ ] train_data = []
for p in train_patients:
    scaled_data = scaler.fit_transform(edf[edf['Patient'] == p]['SpO2'].values.reshape(-1, 1))
    train_data.append(scaled_data)

train_data = np.array(train_data).flatten()
trainX, trainY = create_dataset(train_data, look_back)

[ ] trainX = np.reshape(trainX, (trainX.shape[0], 1, trainX.shape[1]))

[ ] keras.utils.set_random_seed(42)

[ ] model = Sequential()
model.add(LSTM(64, input_shape=(1, look_back)))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

[ ] for p in test_patients:
    test_data = edf[edf['Patient'] == p]['SpO2'].values.astype('float32')
    scaled_test_data = scaler.fit_transform(test_data.reshape(-1, 1))
    testX, _ = create_dataset(scaled_test_data, look_back)

    testX = np.reshape(testX, (testX.shape[0], 1, testX.shape[1]))
    threshold = int(0.75 * len(testX))
    real_values = scaler.inverse_transform(scaled_test_data)

    predictions = []
    print(len(testX)-threshold)
    for i in range(threshold, len(testX), 100):
        last_observations = testX[i:i+1]
        chunk_predictions = []
        for j in range(100):
            if i + j >= len(testX):
                break
            pred = model.predict(last_observations, verbose=0)
            pred_inv = scaler.inverse_transform(pred)
            chunk_predictions.append(pred_inv[0][0])

            last_observations = np.roll(last_observations, -1)
            last_observations[0, 0, -1] = pred[0, 0]

        predictions.extend(chunk_predictions)

    plt.figure(figsize=(100, 60))
    plt.plot(real_values, label='Real Values', color='grey', alpha=0.7)
    plt.plot(np.arange(threshold + look_back, threshold + look_back + len(predictions)), predictions, label='Predictions', color='lime', alpha=0.7)

    plt.title(f'Patient {p}')
    plt.legend()

    plt.show()
mse = mean_squared_error(real_values[threshold+look_back+1:], predictions)
print(f'MSE: {mse}'')
```

Slika 6.36: Postavljanje LSTM mreže, treniranje i predviđanje uz iscrtavanje razine SpO_2 .



Slika 6.37: Predviđanje SpO₂ razina 20 sekundi unaprijed (sivo - ispravne vrijednosti, zeleno - predviđanja).

7. ZAKLJUČAK

Razvoj umjetne inteligencije, preciznije strojnog učenja postavlja pitanje područja njegove primjene. S obzirom na ljudsku ograničenost napredovanja u području medicine, a ujedno i osjetljivost na probleme kojima se bavi, te njihovu rješivost primjenama tehnika umjetne inteligencije, na povezivanje ovih dvaju sfera nije trebalo dugo čekati. Strojno učenje u medicine od svojih začetaka - u kojima je primjena bila ograničena nedoraslošću sklopolavlja. Danas su te potrebe za snagom zadovoljene, stoga svakodnevno svjedočimo uporabi računala koji pogone različite modelle strojnog učenja kao potporu ili potpunu zamjenu liječnicima i dijagnostičarima. Sama apnea spavanja uz otežavanje normalnog svakodnevnog funkciranja organizma ima i opasne dalekosežne posljedice koje je moguće polje razumjeti i spriječiti upotrebnom naprednih modela.

Podaci apneje sna najopsežnije se prikupljaju polisomnografskim studijama u kojima se nad pacijentom kroz noć spavanja provode razna mjerena mnoštvom senzora. Iako su u radu korišteni upravo ovakvi podaci, oni s obzirom na izrazito kontrolirano okruženje podižu pitanje širine primjene modela koji su razvijeni pomoću njih. Najjednostavniji uvid u kvalitetu sna pruža pregled kružnih izmjena fazi sna ispitanika - pravilne izmjene ne-REM i REM fazi. Upravo iz tog razloga u radu je razvijeno nekoliko različitih modela za klasifikaciju fazi sna tokom noći na osnovu signala koji su danas ljudima dostupni svakodnevno - mikrofon za hrkanje, otkucaji srca i razina zasićenosti kisikom danas su lako dostupni kroz razna pametna rješenja poput pametnih satova. Osim izmjena sna, kao dijagnostička metoda često se koriste izračuni mjera poput AHI indeksa - koji govore o kvaliteti sna, ti su proračuni često jako komplikirani i resursno zahtjevni. Primjenom modela strojnog učenja poput konvolucijske mreže pokazalo se kao efektivna zamjena za iste (iako u radu nije razvijen dovoljno precizan model). Glavni problem kod oboljelih sa pojavama apneje i hipopneje su padovi razine kisika u krvi, što je dugoročno najopasniji problem koji se predstavlja kao uzročnik različitih komorbiditeta apneje. Koristeći LSTM mrežu, možemo relativno precizno predviđati takve podatke (buduće razine kisika) unutar uskih vremenskih okvira (nekoliko sekundi ili minuta) što je dovoljno da se pacijenta potakne na buđenje ili pokrene djelovanje sustava CPAP kako bi se izbjeglo oštećenje organa i gušenje pacijenta poradi nedostatka kisika.

LITERATURA

- [1] Wazid, M. Das, A. K. Chamola, V. Park, Y.: Uniting cyber security and machine learning: Advantages, challenges and future research, Volume 8, Issue 3, 2022., s Interneta, <https://doi.org/10.1016/j.icte.2022.04.007>, 27. kolovoza 2023.
- [2] Alzubi, J. Nayyar, A. Kumar, A.: Machine Learning from Theory to Algorithms: An Overview, 2018., s Interneta, <https://doi.org/10.1088/1742-6596/1142/1/012012>, 27. kolovoza 2023.
- [3] Burger, S. V.: Introduction to Machine Learning with R, 2018.
- [4] Bolf, N.: OSVJEŽIMO ZNANJE - Strojno učenje, Vol. 70 No. 9-10, 2021., s Interneta, <https://hrcak.srce.hr/263495>, 27. kolovoza 2023.
- [5] Kaul, V. Enslin, S. Gross, S.A.:History of artificial intelligence in medicine, 2020., s Interneta, <https://doi.org/10.1016/j.gie.2020.06.040>, 27. kolovoza 2023.
- [6] Ujević Andrijić, Ž.: OSVJEŽIMO ZNANJE - Umjetne neuronske mreže, Vol. 68 No. 5-6, 2019., s Interneta, <https://hrcak.srce.hr/file/322233>, 27. kolovoza 2023.
- [7] Philips : Alice PDx, s Interneta: <https://www.philips.hr/healthcare/product/HC1043844/alice-pdx-portable-sleep-diagnostic-system#specifications>, 27. kolovoza 2023.
- [8] Mayo Clinic : Sleep apnea, s Interneta: <https://www.mayoclinic.org/diseases-conditions/sleep-apnea/symptoms-causes/syc-20377631>, 27. kolovoza 2023.
- [9] HeMED : Apneja u snu, s Interneta: <https://www.hemed.hr/Default.aspx?sid=15626>, 27. kolovoza 2023.
- [10] Mayo Clinic : Polysomnography, s Interneta: <https://www.mayoclinic.org/tests-procedures/polysomnography/about/pac-20394877>, 27. kolovoza 2023.
- [11] Specifikacija EDF formata, s Interneta: <https://www.edfplus.info/>, 27. kolovoza 2023.
- [12] Repozitorij Feather projekta, s Interneta: <https://github.com/wesm/feather>, 27. kolovoza 2023.
- [13] Projekt Anaconda, s Interneta: <https://web.archive.org/web/20190302104321/http://docs.anaconda.com/anaconda>, 27. kolovoza 2023.
- [14] Google Research: Google Colaboratory, s Interneta: <https://research.google.com/colaboratory/faq.htm>, 27. kolovoza 2023.

- [15] NumFOCUS: Project Jupyter, s Interneta: <https://numfocus.org/project/project-jupyter>, 27. kolovoza 2023.
- [16] Read The Docs: IPython Notebook Quick Start Guide, s Interneta: https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html, 27. kolovoza 2023.
- [17] IBM: What is random forest?, s Interneta: <https://www.ibm.com/topics/random-forest>, 27. kolovoza 2023.
- [18] Srtuhi, E. R.: Understand Random Forest Algorithms With Examples, 2023., s Interneta: [UnderstandRandomForestAlgorithmsWithExamples](#), 27. kolovoza 2023.
- [19] Bošnjak, M.: Random forests, s Interneta: <http://lis.irb.hr/KDSA2008/presentations/bosnjak.pdf>, 27. kolovoza 2023.
- [20] Verma, S.: Understanding 1D and 3D Convolution Neural Network | Keras, s Interneta: <https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610>, 27. kolovoza 2023.
- [21] Medium: An Overview on Convolutional Neural Networks, s Interneta: <https://medium.com/swlh/an-overview-on-convolutional-neural-networks-ea48e76fb186>, 27. kolovoza 2023.
- [22] Olah, C.: Understanding LSTM Networks, 2015., s Interneta: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>, 27. kolovoza 2023.

POPIS OZNAKA I KRATICA

<i>AHI</i>	indeks apneje i hipopneje
<i>CPAP</i>	Continous positive airway pressure
<i>CSV</i>	Comma Separated Values
<i>EDF</i>	European Data Format
<i>engl</i>	engleski
<i>LSTM</i>	Long Short-Term Memory
<i>ODI</i>	indeks desaturacije krvi kisikom
<i>REM</i>	Rapid Eye Movement
<i>SpO₂</i>	mjera zasićenosti krvi kisikom

SAŽETAK

Sažetak:

Ovaj rad za temu je imao obradu strojnog učenja u medicini i primjenu modela strojnog učenja nad prikupljenim podacima iz područja medicine. Napravljen je prvo općeniti pregled strojnog učenja, povijesti primjena strojnog učenja u medicini, zatim osvrt na medicinski aspekt problema apneje u snu, pregled samih podataka i okruženja te na kraju implementacija modela. Kroz praktični dio rada opisano je prikupljanje i obrada podataka, automatizacija bilježnica koristeći pakete za njihovu parametrizaciju, dok su u posljednjem poglavlju rada, opisani modeli Random Forest, konvolucijske neuronske mreže te LSTM mreže. Također, uz jednostavnu analizu značajki skupa podataka izneseni su rezultati klasifikacije faze sna na osnovu jednog uzorka i na osnovu konteksta s više mreža. Koristeći konvolucijsku neuronsku mrežu izrađen je i model za predviđanje indeksa apneje i hipopneje - AHI čime se može izbjegći kompleksno računanje istog nad razinama kisika. Pomoću LSTM mreže modelirano je kratkotrajno predviđanje razine zasićenosti krvi kisikom.

Ključne riječi:

Umjetna inteligencija, Strojno učenje, Medicina, Apneja, Random forest, Konvolucijske neuronske mreže, LSTM mreže

SUMMARY

Title:

APPLICATION OF MACHINE LEARNING TECHNIQUES TO DATA IN THE FIELD OF MEDICINE

Summary:

The main goal of this master thesis was to showcase the use of machine learning in medicine, and to actually use machine learning models on data from the medical domain. First an overview was made of machine learning in general, then of its usage in the field of medicine, followed by a short walkthrough through the problem of sleep apnea on which the practical problem of the thesis is based on. Following that is a detailed description of the data used in the thesis and how it was gathered, followed also by a theoretical overview of models that will be implemented. First part of the practical work in this thesis is focused on the automation of Jupyter notebook execution for patients using the selected parametrization tools. After that a brief feature analysis was made followed by description of models set up for one sample and contextual classification of sleep phases. Using another convolutional neural network, a model for AHI prediction was made, replacing resource intensive calculations based on oxygen levels. The last part showcases short term predictions of oxygen saturation of blood using a LSTM network in near future frames (couple of seconds).

Key words:

Artificial intelligence, Machine learning, Medicine, Sleep apnea, Random forest, Convolutional neural networks, LSTM networks