

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2040

**PREDVIĐANJE PERCIPIRANE KVALITETE SNA TEMELJEM
OBJEKTIVNIH MJERENJA KVALITETE SNA METODAMA
STROJNOG UČENJA**

Borna Zelić

Zagreb, lipanj 2025.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2040

**PREDVIĐANJE PERCIPIRANE KVALITETE SNA TEMELJEM
OBJEKTIVNIH MJERENJA KVALITETE SNA METODAMA
STROJNOG UČENJA**

Borna Zelić

Zagreb, lipanj 2025.

ZAVRŠNI ZADATAK br. 2040

Pristupnik: **Borna Zelić (0036549534)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

Modul: Računarstvo

Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Predviđanje percipirane kvalitete sna temeljem objektivnih mjerenja kvalitete sna metodama strojnog učenja**

Opis zadatka:

Na kvalitetu sna utječu različiti čimbenici, kao što su dob, cirkadijalni ritam, hormoni i genetika. Percepcija kvalitete sna može se izraziti različitim mjerama, no upitno je koliko je ona predvidljiva u odnosu na objektivno mjerene parametre kvalitete sna (npr. razine hormona, duljina spavanja, prekidi sna). Cilj ovog završnog rada je izrada prediktivnog modela strojnog učenja za percipiranu kvalitetu sna mjerenu na skali od 0 do 21 putem indeksa temeljenog na upitniku "Pittsburgh Sleep Quality Questionnaire Index". Prediktivni model treba se izgraditi temeljem općih informacija o osobi (dob, spol i sl.), objektivno mjerenim značajkama sna, aktivnostima prethodnog dana i varijabilnosti srčanog ritma. Kao skup za učenje modela potrebno je upotrijebiti cijeli ili dio skupa podataka MMASH (Multilevel Monitoring of Activity and Sleep in Healthy people) <https://physionet.org/content/mmash/1.0.0/>. Potrebno je usporediti i odabrati najtočniji prediktivni model strojnog učenja (np. linearna regresija, slučajna šuma za regresiju, XGBoost). U radu je potrebno teorijski opisati biomedicinsku problematiku i metode strojnog učenja. Nakon toga, potrebno je pripremiti podatke, predobraditi ih i izgraditi modele strojnog učenja. Rezultate na zasebnom testnom skupu podataka potrebno je prikazati odgovarajućim mjerama za evaluaciju modela strojnog učenja te ih komentirati. Implementaciju je potrebno napraviti u programskom jeziku po vlastitom izboru, a za izgradnju modela može se u slučaju nedostatka vlastitih sklopovskih resursa koristiti dostupna web rješenja (npr. Google Colab).

Rok za predaju rada: 23. lipnja 2025.

Sadržaj

Uvod	3
1. Medicinska pozadina	4
1.1. Tjelesna aktivnost	4
1.2. Psihološke značajke	5
1.3. Hormonalne značajke	6
2. Tehnička pozadina	7
2.1. Analiza i obrada podataka	7
2.2. Strojno učenje	13
2.2.1. Prenaučenost	13
2.2.2. Učitavanje i obrada podataka	14
2.2.3. Stablo odluke	14
2.2.4. Slučajna šuma	15
2.2.5. Linearna regresija	16
2.2.6. Gaussova naivna Bayesova metoda	17
2.2.7. Ostvareni rezultati i izvoz modela strojnog učenja	17
3. Implementacija	19
3.1. Arhitektura sustava	20
3.1.1. Korisnička strana	20
3.1.2. Poslužiteljska strana	21
4. Zaključak	23
Literatura	24

Sažetak	26
Abstract	27

Uvod

Poznato je da prosječna osoba prespava trećinu svoga života. Nedostatak spavanja i loš san danas su prvi koraci prema lošijem zdravlju. Dok mnogi misle da zbog lošega sna patimo samo u trenutku, u pozadini se mogu događati mnoge loše posljedice po naše zdravlje. Smanjena kvaliteta sna i neispavanost, osim što utječu na raspoloženje [1], mogu dovesti do raznih bolesti krvožilnog i probavnog sustava te pospješiti razvoj nekih vrsta malignih bolesti. Iz navedenih razloga, sve češće se govori o potrebi kvalitetnog sna za zdravlje i fizički razvoj. Kao jedno od rješenja za određivanje kvalitete sna razvijen je indeks "Pittsburgh sleep quality index" kojim se danas numerički može odrediti kvaliteta sna, što primjetno olakšava daljnju psihijatrijsku praksu i istraživanja. U sklopu ovoga rada istražen je "Pittsburgh sleep quality index", analizirana je baza podataka znanstvenoga rada koji je praćenjem raznih fizičkih i psiholoških mjera sna 22 muškaraca odredila „Pittsburgh sleep quality index“ [2], te je napravljen prototip mrežnog sučelja koji na temelju raznih modela strojnoga učenja i unesenih podataka daje predikciju navedenoga indeksa.

1. Medicinska pozadina

Opće je poznato da san može igrati važnu ulogu u metaboličkoj regulaciji, regulaciji emocija, konsolidaciji pamćenja, procesima oporavka mozga i učenju. Unatoč tome, i dalje nije poznato koliko fizička, psihička i hormonalna aktivnost utječu na sam san. Percepciju kvalitete sna moguće je izraziti raznim mjerama, od psihičkih do fizikalnih, zbog čega se pokazuje potreba za pronalaskom jednostavnoga načina za prikaz kvalitete sna na mjerivi način. Zbog toga, ovaj rad pokušava odrediti značajne mjere koje će biti korištene za izučavanje modela strojnoga učenja s ciljem lakoga definiranja kvalitete sna. Istraživanje korišteno u nastavku rada prikazuje kvalitetu sna kroz 44 značajke, a kao mjerilo kvalitete sna koristi "Pittsburg Sleep Quality Index" (dalje PSQI). PSQI upitnik je upitnik za samoprocjenu koji određuje kvalitetu i poremećaje sna [3]. PSQI na temelju 19 različitih pitanja svrstanih u 7 kategorija (subjektivna kvaliteta sna, latencija spavanja, trajanje spavanja, uobičajena učinkovitost spavanja, poremećaji spavanja, upotreba lijekova za spavanje i dnevna disfunkcija) daje indeks koji poprima vrijednosti od jedan do dvadeset i jedan. Svi indeksi manji od šest ukazuju na dobru kvalitetu sna. U istraživanju je sudjelovalo dvadeset i dva zdrava muškarca kojima su se 24 sata bilježili otkucaji srca, svi pokreti tijela s pomoću triaksijalnog akcelerometra, tjelesna aktivnost, psihološke karakteristike te lučenje hormona.

1.1. Tjelesna aktivnost

Mjere tjelesne aktivnosti mogu se podijeliti na četiri kategorije: otkucaji srca, pokreti zabilježeni triaksijalnim akcelerometrom, aktivnosti te informacije tijekom spavanja. Zabilježen je svaki otkucaj srca, vrijeme i datum kad se dogodio i vremenski razmak između dva uzastopna otkucaja srca. Pokreti zabilježeni triaksijalnim akcelerometrom zabilježeni su kroz 12 kategorija, od kojih se ističu x, y i z koordinate, broj koraka te podatci

inklometra koji opisuju poziciju osobe (stajanje, sjedenje i ležanje). Aktivnost je prikazana kao indeks u kojemu su brojčane vrijednosti mapirane na određene aktivnosti, npr. sjedenje, ležanje, lagana kretnja, jedenje, pušenje i sl. Informacije tijekom spavanja iznimno su bitne za ovaj rad, a prikazane su značajkama kao vrijeme početka i kraja spavanja, ukupno vrijeme spavanja i ukupno vrijeme provedeno u krevetu, broj buđenja i količina vremena provedenog budnim nakon prvog buđenja te indeksi o količini vremena provedenog bez i s kretanjem tijekom spavanja.

Podatci o tjelesnoj aktivnosti od iznimne su važnosti jer detaljno prikazuju sve što je prethodilo spavanju i iz tih podataka možemo donositi zaključke kako dnevna tako i aktivnost tijekom spavanja utječu na san.

1.2. Psihološke značajke

Psihološke značajke izražene su kroz različite vrste indeksa. U znanstvenome istraživanju ponuđeni su Morningness-Eveningness Questionnaire indeks, State-Trait Anxiety Inventory indeks, Behavioural avoidance/inhibition indeks, Daily Stress Inventory indeks i Positive and Negative Affect Schedule indeks. Oni prikazuju psihološko stanje osobe za vrijeme istraživanja i daju nam uvid u dnevne navike ispitanika. Morningness-Eveningness Questionnaire (MEQ) daje vrijednosti od 16 do 86 gdje vrijednosti veće od 59 označavaju jutarnji tip osobe, vrijednosti manje od 41 noćni tip osobe, a sve vrijednosti između označavaju srednji tip. U upitniku se odgovara na pitanja o fizičkim i psihičkim naporima kroz dan gdje ispitanik sam mora procijeniti svoje sposobnosti snalaženja s istima. Upitnik anksioznosti kao stanja i osobine ličnosti (STAI) namijenjen je mjerenju anksioznosti u odraslih osoba [3]. Rezultati se kreću u rasponu od 20 do 80, gdje rezultati manji od 31 ukazuju na nepostojanje anksioznosti, rezultati do 49 ukazuju na postojanje anksioznosti, dok vrijednosti od 50 i više ukazuju na visok stupanj anksioznosti. Upitnik se dijeli u dvije ljestvice, od kojih jedna referencira na prolazna emocionalna stanja, dok druga govori o mjeri anksioznosti kao osobini ličnosti. Behavioural avoidance system (BAS) i Behavioural inhibition system (BIS) skale su koje se koriste za određivanje sklonosti anksioznosti, depresiji i maničnom ponašanju kod odraslih [4]. Daily Stress Inventory vrijednost je mjera uvedena kako bi se istraživačima i kliničarima pružio psihometrijski ispravan instrument samoprocjene za dnevnu procjenu izvora i individu-

aliziranog utjecaja relativno manjih stresnih događaja [5]. Ispitanik u upitniku opisuje događaje u danu te im zadaje vrijednost količine stresa koju su izazvale. Na kraju, upitnik daje vrijednosti od 0 do 406 gdje veće vrijednosti predstavljaju veću količinu stresa u danu. Positive and Negative Affect Schedule (PANAS) je skala koja se može koristiti za mjerenje afekta stanja, dispozicijskog ili osobinskog afekta, emocionalnih fluktuacija tijekom određenog razdoblja ili emocionalnih reakcija na događaje [6]. PANAS daje vrijednosti od 5 do 50 gdje veće vrijednosti daju veći stupanj doživljenog afekta u vremenskom okviru.

1.3. Hormonalne značajke

Hormoni su kemijski glasnici u tijelu svih živih bića koji reguliraju i koordiniraju različite funkcije u tijelu. Regulacija hormona iznimno je bitna i već male razlike mogu znatno utjecati na osobe. U ponuđenom znanstvenom radu hormoni melatonin i kortizol bili su očitani iz uzoraka sline. Kortizol je hormon koji se izlučuje iz nadbubrežnih žlijezda i kontrolira potrošnju pohranjene glukoze u organizmu, regulira tlak te znatno utječe na san. Pozant je kao hormon stresa koji je uključen u odgovor na fizički i/ili emocionalni stres [7]. Melatonin je hormon koji je danas najpoznatiji po tome što regulira san, no ima i razne druge važne uloge poput suzbijanja rasta karcinoma [8]. Razina melatonina u tijelu najviša je navečer te je odgovoran za osjećaj umora, dok je ujutro nakon buđenja najmanja. Prirodna proizvodnja melatonina raste kako je okolina tamnija i kako dan odmiče, zbog čega se često preporučuje izbjegavanje plavoga svjetla prije spavanja i zatamnjenje sobe u kojoj se spava.

2. Tehnička pozadina

2.1. Analiza i obrada podataka

U današnje doba postoji mnogo načina za prikupljanje, spremanje i analizu podataka. Jedan od najčešće korištenih oblika prikazivanja podataka je "Comma-separated values" (dalje CSV) format. Podatci takvoga formata prikazani su kao podatci razdijeljeni zarezom, gdje prvi redak tablice predstavlja imena značajki tablice, a svaki sljedeći redak tipično predstavlja jedan podatkovni zapis. Takva je vrsta zapisa spremljena u .csv datoteke, a krasi ga prenosivost i jednostavnost. Vrijednosti dobivene u danoj bazi podataka bile su raspršene u nekoliko raznih CSV datoteka, te je za spajanje istih napravljena jednostavna Python skripta. Nakon učitavanja i spajanja svih potrebnih CSV datoteka, proveden je proces analize i selekcije podataka. S obzirom na relativno veliku količinu podataka za relativno malu količinu različitih uzoraka, ključno je bilo korisno iskoristiti sve uzorke. Za analizu podataka i biranje korisnih informacija odabran je programski jezik R. Zbog male količine dostupnih informacija, na samome je početku trebalo ukloniti neke značajke kako bi se dobilo dovoljno stupnjeva slobode za računanje linearne regresije kojom će se utvrditi postoji li zavisnost između značajki koja bi kasnije mogla utjecati na rezultate te odstupanje prave vrijednosti od predviđene vrijednosti. Ovaj dio obrade podataka ključan je kako bi se podatci pripremili za učenje i testiranje modela strojnoga učenja.

analiza

2025-06-15

Na početku je potrebno učitati podatke iz .csv datoteke. Nakon što su podatci učitani, potrebno je učitati sve potrebne datoteke s paketima u R-u koje će kasnije biti korištene. Na kraju, potrebno je maknuti određene značajke. Neke značajke se miču jer postoji malo različitih podataka i previše varijabli što otežava izgradnju točnog modela strojnog učenja.

```
original_dataset = read.csv("merged.csv")
library(tidyverse)

library(here)

library(ggplot2)
library(corrplot)

library(caret)

library(car)

library(ggcorrplot)

dataset <- original_dataset[, !colnames(original_dataset) %in% c("Gender",
"Efficiency", "In.Bed.Date", "In.Bed.Time", "Out.Bed.Date", "Out.Bed.Time",
"Onset.Date", "Onset.Time", "X")]
set.seed(123)
```

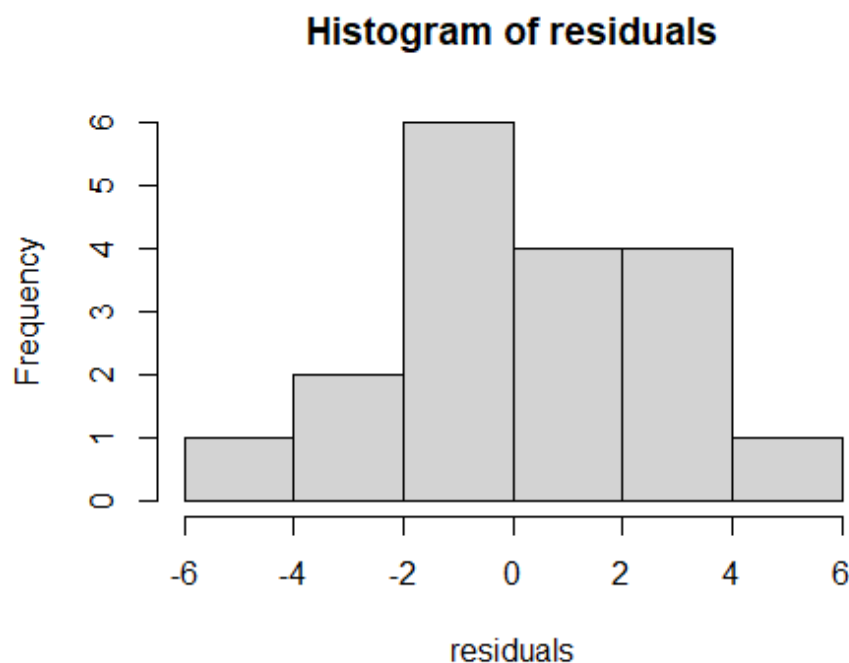
Značajke smo micali kako bi pri pravljenju linearne regresije imali dovoljno stupnjeva sloboda. Na histogramu (Slika 1.) se vidi raspodjela reziduala koja nije idealna, no nazire se zvonoliki izgled stupaca. Rezidualne prikazujemo grafički kako bi lakše uočili njihovu raspodjelu. Rezidualne su bitne jer nam prikazuju koliko je odstupanje između predviđene i prave vrijednosti koju tražimo.

```
lm_model <- lm(data = dataset)
residuals <- lm_model$residuals
summary(lm_model)
```

```
##
## Call:
## lm(data = dataset)
##
## Residuals:
##      1      2      3      4      5      6      7      8      9
## 2.0832 -2.9837  2.5188 -1.1155 -5.5780  0.4745 -0.1553 -0.4822  0.1035
## 10
## 2.0832 -2.9837  2.5188 -1.1155 -5.5780  0.4745 -0.1553 -0.4822  0.1035
## 11      12      13      14      15      16      17      18
## 2.7162  2.3932 -1.6639 -1.8374 -3.5738 -0.1699  1.5646  5.6019
##
```

```
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -6.107e+01  1.310e+02  -0.466   0.665
## Height         7.762e-01  8.932e-01   0.869   0.434
## Age          -1.009e-01  4.293e-01  -0.235   0.826
## Latency        2.127e+00  2.532e+00   0.840   0.448
## Total.Minutes.in.Bed -9.509e-02  2.801e-01  -0.339   0.751
## Total.Sleep.Time..TST.  5.196e-02  2.593e-01   0.200   0.851
## Wake.After.Sleep.Onset..WASO.      NA         NA      NA      NA
## Number.of.Awakening  8.438e-02  8.084e-01   0.104   0.922
## Average.Awakening.Length  9.657e-01  3.743e+00   0.258   0.809
## Movement.Index    5.767e-01  1.302e+00   0.443   0.681
## Fragmentation.Index -2.938e-01  3.365e-01  -0.873   0.432
## Sleep.Fragmentation.Index      NA         NA      NA      NA
## before.sleep.Cortisol.NORM  4.094e+01  8.454e+01   0.484   0.653
## before.sleep.Melatonin.NORM  1.728e+05  9.957e+08   0.000   1.000
## wake.up.Cortisol.NORM    3.597e+01  1.012e+02   0.355   0.740
## wake.up.Melatonin.NORM    6.007e+07  8.576e+08   0.070   0.948
##
## Residual standard error: 5.439 on 4 degrees of freedom
## Multiple R-squared:  0.9109, Adjusted R-squared:  0.6212
## F-statistic: 3.145 on 13 and 4 DF,  p-value: 0.1391
```

`hist(residuals)`

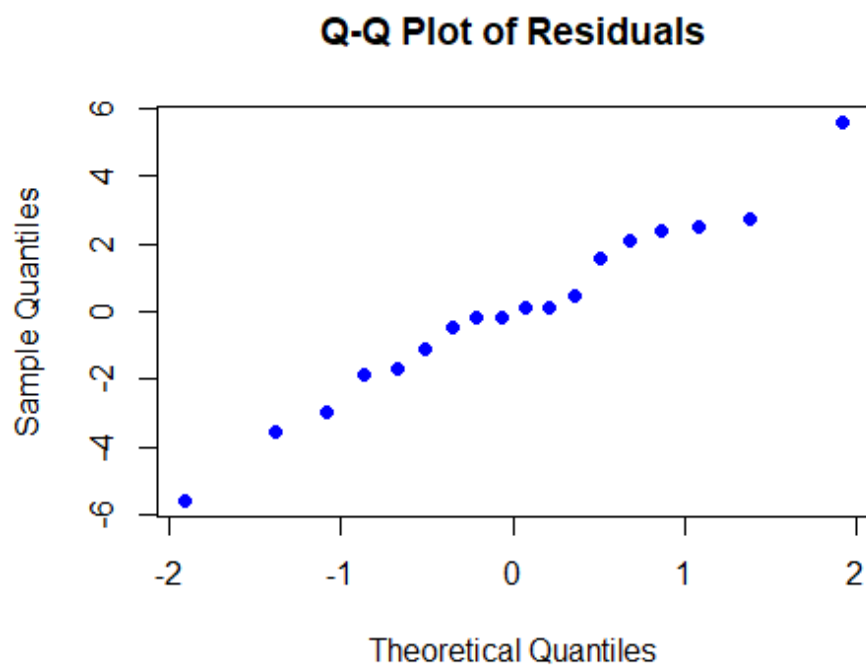


Slika 1. Histogram reziduala

```
plot(hist(residuals))
```

Na sljedećem dijagramu (Slika 2.) također vidimo raspoređenost reziduala na kojemu se nazire linearnost.

```
plot.new
## function ()
## {
##   for (fun in getHook("before.plot.new")) {
##     if (is.character(fun))
##       fun <- get(fun)
##     try(fun())
##   }
##   .External2(C_plot_new)
##   grDevices::recordPalette()
##   for (fun in getHook("plot.new")) {
##     if (is.character(fun))
##       fun <- get(fun)
##     try(fun())
##   }
##   invisible()
## }
## <bytecode: 0x0000021bc3aaa900>
## <environment: namespace:graphics>
qqnorm(residuals, main = "Q-Q Plot of Residuals", pch = 19, col = "blue")
```



Slika 2. Q-Q plot reziduala

Zatim, na ispisu konzole možemo vidjeti međusobnu korelaciju parova stupaca. U slučaju da je korelacija pre visoka, značajke su zavisne te nema potrebe da koristimo oba iz para.

```
dataset_without_target <- dataset[, !colnames(dataset) %in% "Pittsburgh"]
numeric_data <- data.frame(model.matrix(~ . - 1, data =
dataset_without_target))
corr_matrix <- round(cor(numeric_data), 2)
corr_pairs <- as.data.frame(as.table(corr_matrix))
high_corr_pairs <- subset(corr_pairs, abs(Freq) > 0.9 & Var1 != Var2)
print(high_corr_pairs)
```

```
##              Var1              Var2 Freq
## 70    Total.Sleep.Time..TST.    Total.Minutes.in.Bed 0.93
## 85    Total.Minutes.in.Bed    Total.Sleep.Time..TST. 0.93
## 172 Sleep.Fragmentation.Index    Fragmentation.Index 0.93
## 187    Fragmentation.Index Sleep.Fragmentation.Index 0.93
```

```
summary(lm_model)
```

```
##
## Call:
## lm(data = dataset)
##
## Residuals:
##      1      2      3      4      5      6      7      8      9
## 10
## 2.0832 -2.9837  2.5188 -1.1155 -5.5780  0.4745 -0.1553 -0.4822  0.1035
## 0.1038
##     11     12     13     14     15     16     17     18
## 2.7162  2.3932 -1.6639 -1.8374 -3.5738 -0.1699  1.5646  5.6019
##
## Coefficients: (2 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -6.107e+01  1.310e+02  -0.466   0.665
## Height         7.762e-01   8.932e-01   0.869   0.434
## Age           -1.009e-01   4.293e-01  -0.235   0.826
## Latency        2.127e+00   2.532e+00   0.840   0.448
## Total.Minutes.in.Bed -9.509e-02  2.801e-01  -0.339   0.751
## Total.Sleep.Time..TST.  5.196e-02  2.593e-01   0.200   0.851
## Wake.After.Sleep.Onset..WASO.      NA         NA      NA      NA
## Number.of.Awakenings  8.438e-02  8.084e-01   0.104   0.922
## Average.Awakening.Length  9.657e-01  3.743e+00   0.258   0.809
## Movement.Index    5.767e-01  1.302e+00   0.443   0.681
## Fragmentation.Index -2.938e-01  3.365e-01  -0.873   0.432
## Sleep.Fragmentation.Index      NA         NA      NA      NA
## before.sleep.Cortisol.NORM  4.094e+01  8.454e+01   0.484   0.653
## before.sleep.Melatonin.NORM  1.728e+05  9.957e+08   0.000   1.000
## wake.up.Cortisol.NORM    3.597e+01  1.012e+02   0.355   0.740
## wake.up.Melatonin.NORM    6.007e+07  8.576e+08   0.070   0.948
##
## Residual standard error: 5.439 on 4 degrees of freedom
```

```
## Multiple R-squared:  0.9109, Adjusted R-squared:  0.6212
## F-statistic: 3.145 on 13 and 4 DF,  p-value: 0.1391
```

Za kraj, izbaciti ćemo varijable s velikom frekvencijom korelacije i tražit ćemo signifikantne varijable na koje ćemo se kasnije fokusirati pri učenju modela strojnog učenja, dok ćemo ostale značajke odbaciti jer nam ne pridonose preciznosti modela. Također, višak značajki može negativno utjecati na model strojnoga učenja jer će dodavati šum.

```
dataset_without_target <- dataset_without_target[,
!colnames(dataset_without_target) %in% c("Total.Minutes.in.Bed",
"Fragmentation.Index")]
lm_model <- lm(data = dataset_without_target)
significant_vars <-
names(coef(lm_model))[which(summary(lm_model)$coefficients < 1)]
significant_vars <- setdiff(significant_vars, "(Intercept)")
significant_vars <- significant_vars[1:8]
print(significant_vars)

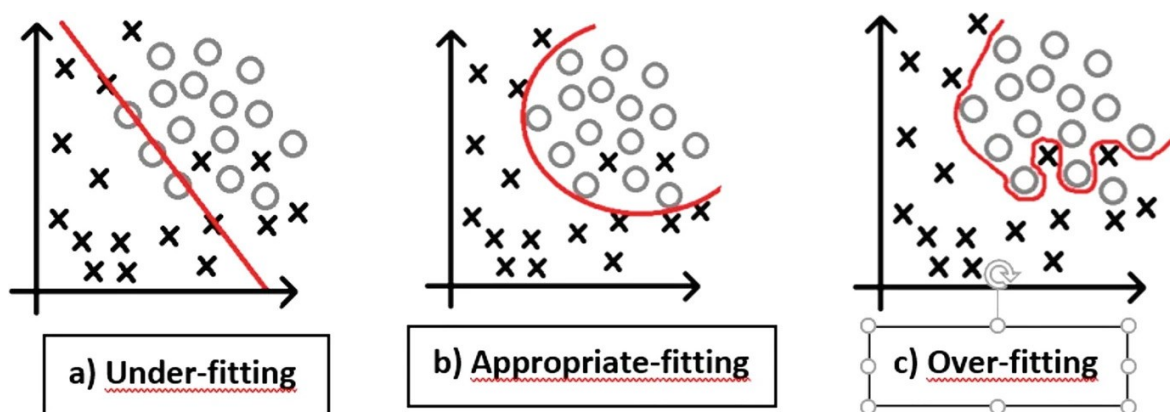
## [1] "Height" "Age"
## [3] "Total.Sleep.Time..TST." "Wake.After.Sleep.Onset..WASO."
## [5] "Number.of.Awakening" "Average.Awakening.Length"
## [7] "Movement.Index" "Sleep.Fragmentation.Index"
```

2.2. Strojno učenje

Strojno učenje grana je umjetne inteligencije koja omogućuje računalima da na temelju podataka za učenje uče o nekome problemu [9]. Jednom kada je model strojnoga učenja dobro naučen o zadanome problemu, možemo ga koristiti da, na temelju već viđenih vrijednosti, iz novo dobivenih vrijednosti donosi zaključke. Modeli strojnoga učenja dijele se na tri velike skupine: nadzirano učenje, nenadzirano učenje i podržano učenje. Karakteristika nadziranoga strojnoga učenja u odnosu na ostale vrste je što se uči na skupu podataka koji pruža ulazne vrijednosti, ali i točne izlazne vrijednosti. To ujedno takvu vrstu strojnoga učenja čini najjednostavnijom za implementaciju i razumijevanje te se može primijeniti na razne probleme, pa tako i na onaj u ovome radu. Za modele strojnoga učenja bazirane na nadziranome učenju potrebno je podatke za učenje podijeliti na dva, odnosno tri dijela. Obavezni dijelovi moraju biti podatci za učenje i podatci za testiranje, dok su podatci za validaciju opcionalni. Podatci se dijele na te skupove kako bi se spriječila prenaučenost.

2.2.1. Prenaučenost

Prenaučeni model strojnoga učenja (engl. overfitting) onaj je model koji se zbog velike količine podataka za testiranje prilagodio istima i za već viđene podatke daje vrlo dobru točnost (Slika 2.1.c)), no za neviđene podatke daje lošu točnost. Kako bi se takvo što



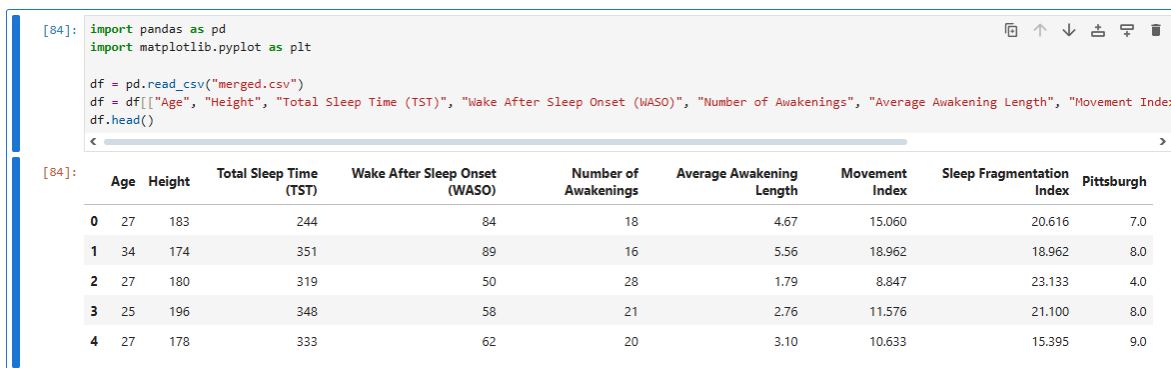
Slika 2.1. Prenaučenost modela strojnog učenja
[10]

izbjeglo, ostavlja se određena količina podataka na kojima se model ne uči, već se koriste da se na njima izračuna točnost izlaza naučenog modela. U slučaju niske točnosti za neviđeni skup podataka, primjenjuju se neke mjere poput podrezivanja i eliminacijom

značajki koji ne doprinose točnosti modela.

2.2.2. Učitavanje i obrada podataka

Model strojnoga učenja napravljen je pomoću Pythonovog mrežnog sučelja Jupyter. U Jupyter notebooku, podatci su učitani koristeći pandas biblioteku pandas (slika 2.2.) i odabrane su one značajke koje su u prethodnome poglavlju utvrđene kao značajne. Na-



```
[84]: import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("merged.csv")
df = df[["Age", "Height", "Total Sleep Time (TST)", "Wake After Sleep Onset (WASO)", "Number of Awakenings", "Average Awakening Length", "Movement Index", "Sleep Fragmentation Index", "Pittsburgh"]]
df.head()
```

	Age	Height	Total Sleep Time (TST)	Wake After Sleep Onset (WASO)	Number of Awakenings	Average Awakening Length	Movement Index	Sleep Fragmentation Index	Pittsburgh
0	27	183	244	84	18	4.67	15.060	20.616	7.0
1	34	174	351	89	16	5.56	18.962	18.962	8.0
2	27	180	319	50	28	1.79	8.847	23.133	4.0
3	25	196	348	58	21	2.76	11.576	21.100	8.0
4	27	178	333	62	20	3.10	10.633	15.395	9.0

Slika 2.2. Obrada podataka u Jupyter notebooku

kon što su podatci uneseni, koristi se biblioteka sklearn [11] za podjelu podataka na skup podataka za učenje i skup podataka za testiranje te isprobavanje raznih vrsta nadziranog strojnoga učenja. Raspodjela skupa podataka prikazana je na slici 2.3. S podijeljenim

```
[10]: x = df.drop("Pittsburgh", axis = 1)
y = df["Pittsburgh"]

[66]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    x, y, test_size=0.33, random_state=1011
)
```

Slika 2.3. Raspodjela skupa podataka

podacima i uz pomoć biblioteke sklearn stvaraju se različite vrste modela nadziranog strojnoga učenja. Kao prvi model odabran je onaj najjednostavniji, stablo odluke.

2.2.3. Stablo odluke

Stablo odluke smatra se jednostavnim modelom jer ne zahtjeva veliku količinu provjera i računanja pri predikciji novih rezultata. Jedna od implementacija stabla odluke je algoritam ID3 koji pomoću mjere neodređenosti podataka pokušava podijeliti podatke u grane. Takvu podjelu zovemo podjela u odnosu na informacijsku dobit, koju računamo

```

from sklearn.tree import DecisionTreeRegressor
dec_tree_reg = DecisionTreeRegressor(random_state=0)
dec_tree_reg.fit(X_train, y_train.values)
y_pred = dec_tree_reg.predict(X_test)
error = np.sqrt(mean_squared_error(y_test, y_pred))
print(error)
4.041451884327381

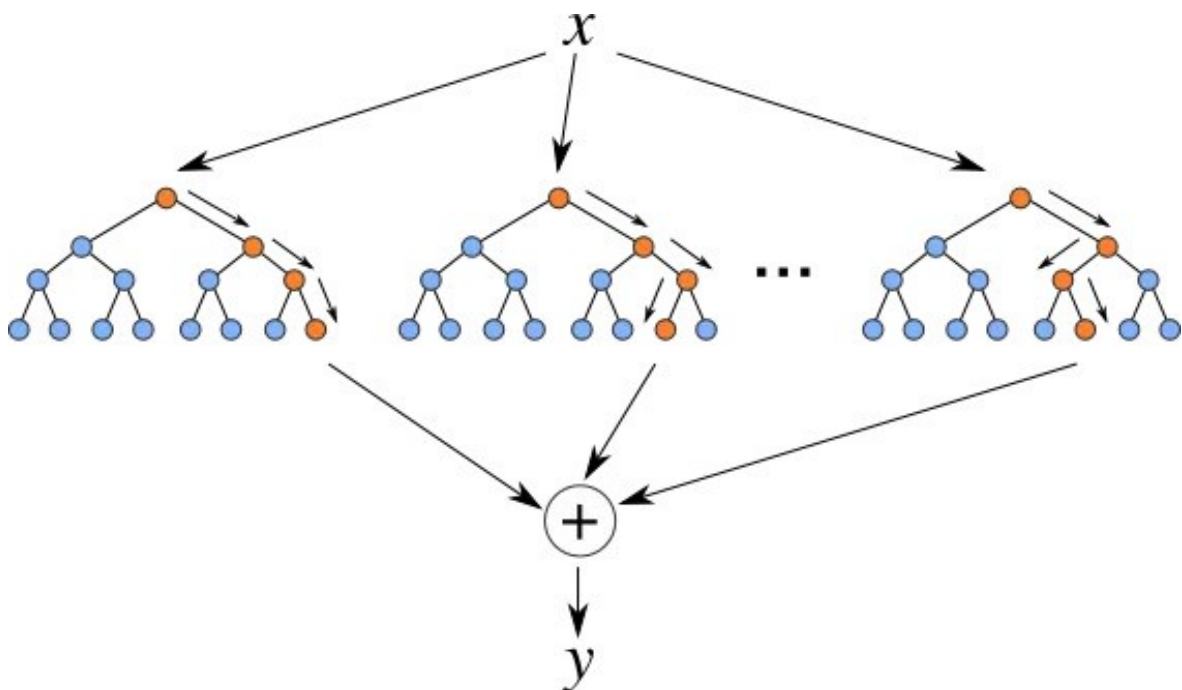
```

Slika 2.4. Model strojnog učenja metodom stabla odluke

oduzimanjem sume umnožaka mjera podataka u skupu i entropije podijeljenog skupa od entropije početnog skupa. U ovome radu koristi se algoritam `DecisionTreeRegressor` iz `sklearn`, a kod je prikazan na slici 2.4. Taj algoritam implementiran je kao "Classification And Regression Tree" algoritam (dalje CART). CART algoritam elemente razvrstava elemente po Gini-indeksu koji se računa kao suma kvadriranih omjera klasifikacija od jedan. Značajka s najvećim Gini-indeksom bira se kao značajka grananja i algoritam se ponavlja dok se ne odrede sve vrijednosti ili dok se ne dostigne zadana dubina odrezivanja.

2.2.4. Slučajna šuma

Kao jednostavno poboljšanje u odnosu na stablo odluke nudi se Slučajna šuma. Slučajna



Slika 2.5. Model strojnog učenja metodom linearne regresije [12]

šuma sastoji se od veće količine različitih stabla odluke. Svako stablo odluke gradi se na svome podskupu podataka, a taj se podskup računa algoritmom "bootstrap aggregation". "Bootstrap aggregation" je algoritam koji novi skup podataka radi slučajnim odabirom

podataka iz originalnoga skupa podataka, češće birajući podatke koji se češće pojavljuju u originalnom skupu podataka, što omogućava gotovo neograničenu količinu skupova podataka. Svako stablo se uči CART algoritmom na svojem skupu podataka, uz slučajan izbor podskupa od ukupnog broja značajki za grananje u svakom čvoru, a za predviđanje slučajna šuma računa izlazne vrijednosti za sva svoja stabla i zatim uzima prosijek tih vrijednosti (slika 2.5.). Slučajna šuma tako pokušava riješiti problem prenaučivosti koji je karakterističan za jedno stablo odluke, no i dalje pokušava zadržati jednostavnost izgradnje modela. 2.6.

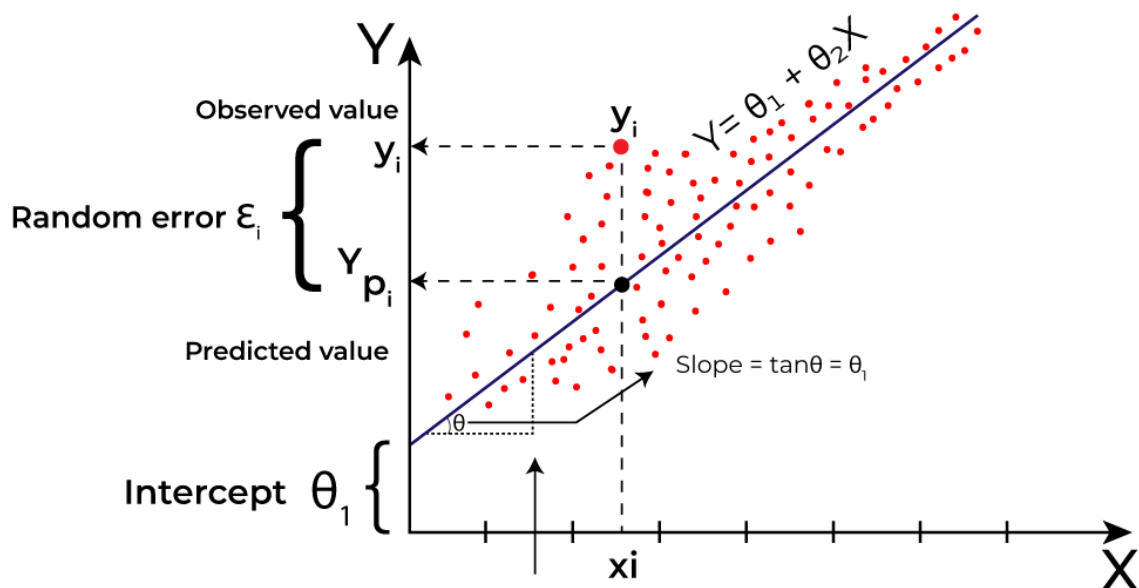
```
from sklearn.ensemble import RandomForestRegressor
random_forest_reg = RandomForestRegressor(random_state=0)
random_forest_reg.fit(X_train, y_train.values)
y_pred = random_forest_reg.predict(X_test)
error = np.sqrt(mean_squared_error(y_test, y_pred))
print(error)
```

2.602447565914313

Slika 2.6. Model strojnog učenja metodom slučajne šume

2.2.5. Linearna regresija

Daljnja opcija može biti linearna regresija koja je jedan od najčešćih oblika strojnoga učenja. Izgled modela linearne regresije prikazan je na slici 2.7. Linearna regresija funk-



Slika 2.7. Model strojnog učenja metodom linearne regresije [13]

cionira tako što pretpostavlja da postoji linearan odnos između ulaza i izlaza te učenjem

pokušava odrediti koji je odnos između svake ulazne vrijednosti i izlazne vrijednosti. Model linearne regresije se uči tako što mijenjajući vrijednosti nagiba pravca pokušava smanjiti pogreška, često implementirane kao sume kvadrata reziduala. U odnosu na obično stablo odluke, linearna regresija nije toliko podložna prenaučivosti te bolje radi kad je veći broj ulaznih vrijednosti u odnosu na stablo odluke [14], no na ponuđeni skup podataka nije pouzdana.

```
from sklearn.linear_model import LinearRegression
linear_reg = LinearRegression()
linear_reg.fit(X_train, y_train.values)
y_pred = linear_reg.predict(X_test)
from sklearn.metrics import mean_squared_error, mean_absolute_error
import numpy as np
error = np.sqrt(mean_squared_error(y_test, y_pred))
print(error)
4.229063365745646
```

Slika 2.8. Model strojnog učenja metodom linearne regresije

2.2.6. Gaussova naivna Bayesova metoda

Kao zadnja metoda istražena u sklopu ovoga rada prikazuje se Gaussov naivni Bayesov klasifikator. Riječ je o jednostavnom, ali učinkovitom modelu strojnog učenja koji se temelji na Bayesovom teoremu i pretpostavci uvjetne neovisnosti varijabli. U ovoj varijanti pretpostavlja se da se kontinuirane značajke unutar svake klase distribuiraju prema Gaussovoj (normalnoj) razdiobi. Parametri modela, poput srednje vrijednosti i standardne devijacije za svaku značajku unutar svake klase, procjenjuju se na temelju podataka iz skupa za učenje. Ova metoda poznata je po svojoj brzini i robusnosti te, iako se sve rjeđe koristi u složenim sustavima zbog snažnih pretpostavki, i dalje pronalazi svoju primjenu u zadacima koji zahtijevaju brzi i probabilistički odgovor. Rezultati takvoga modela prikazan su na slici 2.9.

```
from sklearn.naive_bayes import GaussianNB
GaussianNB_reg = GaussianNB()
GaussianNB_reg.fit(X_train, y_train.values)
y_pred = GaussianNB_reg.predict(X_test)
error = np.sqrt(mean_squared_error(y_test, y_pred))
print(error)
3.0550504633038935
```

Slika 2.9. Model strojnog učenja Gaussovim naivnim Bayesovim kvantifikatorom

2.2.7. Ostvareni rezultati i izvoz modela strojnog učenja

Tablični prikaz ostvarenih korijena kvadriranih srednjih vrijednosti pogrešaka (engl. root mean square error) za svaki model prikazan je tablicom 2.1. Na tablici je vidljivo kako

model slučajne šume ostvaruje najmanju pogrešku. Zbog najboljih rezultata na ponu-

Model strojnog učenja	RMSE
Stablo odluke	4.041451884327381
Slučajna šuma	2.602447565914313
Linearna regresija	4.229063365745646
Gaussov naivni Bayesov kvantifikator	3.0550504633038935

Tablica 2.1. Tablica korijena kvadratnih srednjih vrijednosti pogrešaka

đenim podacima i zbog svoje otpornosti na prenaučenosť u nastavku rada korišten je model slučajna šuma za predviđanje na temelju strojnoga učenja. Taj je model pomoću

```
import pickle
with open('saved_steps.pkl', 'wb') as file:
    pickle.dump({"model":random_forest_reg}, file)
```

Slika 2.10. Pohrana modela strojnog učenja u .pkl datoteku

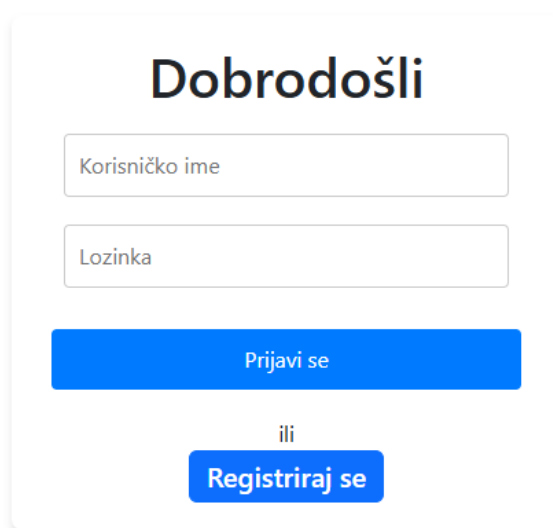
biblioteke pickle pohranjen iz Jupyter notebook-a u .pkl datoteku (slika 2.10.) koja se otvara pri pokretanju web aplikacije te se iz nje uzima objekt u kojem su pohranjeni svi podatci o modelu (slika 2.11.).

```
with open('saved_steps.pkl', 'rb') as file:
    data = pickle.load(file)
    regressor_loaded = data["model"]
```

Slika 2.11. Dobavljanje modela strojnog učenja iz .pkl datoteke

3. Implementacija

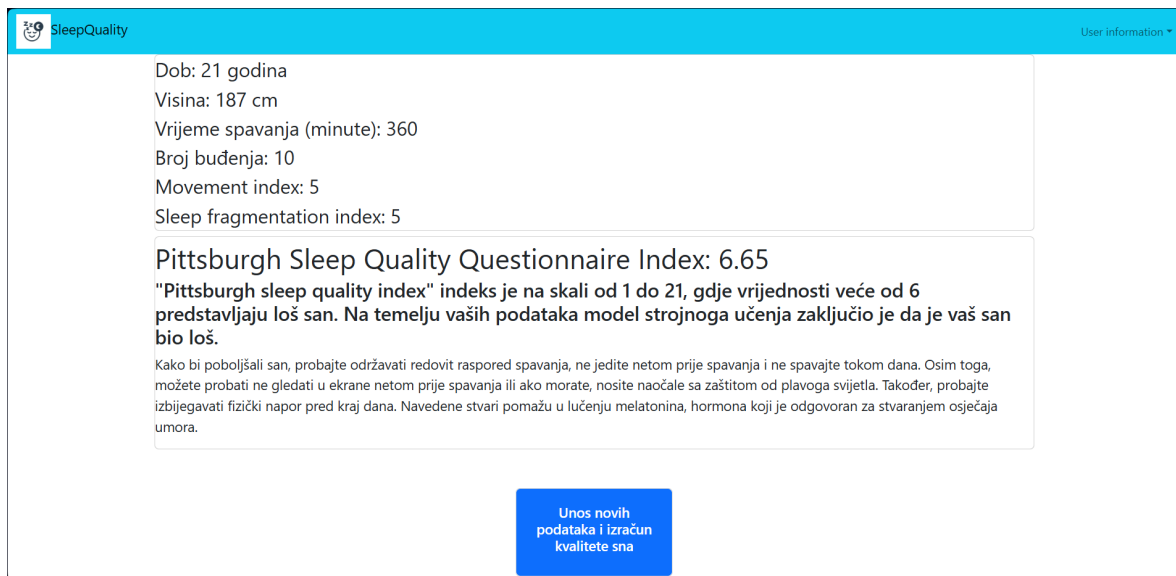
Kako bi korisnik mogao koristiti mrežno sučelje, potrebno se prvo prijaviti. Početno sučelje od korisnika traži korisničko ime i lozinku, slika 3.1., a u slučaju da korisnik još nema profil, može se registrirati. Nakon što se korisnik uspješno prijavi u aplikaciju,



The image shows a login form with a white background and rounded corners. At the top, the title "Dobrodošli" is centered in a large, bold, black font. Below the title, there are two input fields: the first is labeled "Korisničko ime" and the second is labeled "Lozinka". Both fields have a light gray border. Below the input fields, there is a blue button with the text "Prijavi se" in white. Below this button, the word "ili" is centered in a small, gray font. At the bottom, there is another blue button with the text "Registriraj se" in white.

Slika 3.1. Prijava u mrežno sučelje

preusmjeren je na početnu stranicu aplikacije. Na početnoj stranici aplikacije korisnik može vidjeti neke generalne savjete za bolji i lakši san, može odabrati opciju unosa podataka o snu i predikcije "Pittsburgh sleep quality index", uređivanje osobnih informacija te odjavljivanje iz aplikacije. Primjer predikcije indeksa za unesene podatke korisnika prikazan je na slici 3.2.



Slika 3.2. Stranica za predikciju vrijednosti "Pittsburgh sleep quality index"

3.1. Arhitektura sustava

3.1.1. Korisnička strana

Korisnička strana aplikacije pisana je u jeziku JavaScript pomoću radnoga okvira Vite-React. Ovaj radni okvir odabran je zbog svoje popularnosti i jednostavnosti, a glavni zadatak mu je pružiti dio funkcionalnosti vezan za Pogled u okviru obrasca Model-Pogled-Nadglednik. Pogodan je za razvoj korisničke strane aplikacije zbog mogućnosti kombi-

```

11 function Home() {
12   const [loading, setloading] = useState("")
13   const [data, setData] = useState({})
14
15   useEffect(() => {
16     getinfo();
17   }, []);
18
19   const getinfo = async (e) => {
20     setloading(true);
21     try {
22       const res = await api.get("/api/user/info/").catch((err) => console.log(err))
23       setData(res.data)
24     } catch (error) {
25       console.log(error)
26     } finally {
27       setloading(false)
28     }
29   }
30   return (
31     <div id="main_div">
32       <Navbar route="/api/user/info/" method="userinfo"/>
33       <div class="d-flex justify-content-center align-items-center flex-column">
34         <div class="card w-75 text-left fs-2">
35           Welcome to SleepQuality, {data["first_name"]}
36         </div>
37       </div>
38       <div class="card w-75 text-left fs-3 text-info card-title">...
39     </div>
40     <div class="d-flex justify-content-center align-items-center flex-column m-5 w-75">
41       You, 14 hours ago • Uncommitted changes ...
42     </div>
43   </div>
44 )
45 }
46
47 export default Home;

```

Slika 3.3. Primjer Reactove komponente s Bootstrapom

niranja raznih komponenti u veće cjeline od kojih se na kraju gradi stranica. Uz React, za stiliziranje aplikacije koristi se Bootstrap. Bootstrap je CSS-ov radni okvir razvijen u sklopu Twittera, a pruža veliku količinu raznih gotovih rješenja za razne elemente poput padajućih izbornika, gumba, navigacijskih traka i slično. , slike 3.3.

3.1.2. Poslužiteljska strana

Poslužiteljska strana aplikacije pisana je jezikom Python u radnome okviru Django. Django je najpoznatiji Pythonov radni okvir za rad na poslužiteljskoj strani, no u zadnje vrijeme se manje koristi. Kao i sam jezik, veoma je slobodan i dosta nudi pri samoj inicijalizaciji projekta. Aplikacija pisana u Django zapravo se sastoji od manjih aplikacija koje se pokušavaju odvojiti jedna od druge kako bi se postigla lakša prenosivost. Django odmah dolazi s podrškom za bazu podataka SQLite s kojom komunicira putem *Object Relational Mapping* (ORM-a) (slika 3.4.). ORM omogućava korisniku jednostavno baratanje podacima i ne zahtijeva veliku brigu te odrađuje Model dio Model-Pogled-Upravljač obrasca. Upravljački dio Model-Pogled-Upravljač obrasca Django obavlja u obliku objekata ime-

```
7 class SleepData(models.Model): 3 usages
8     total_minutes_in_bed = FloatField()
9     total_sleep_time = FloatField()
10    wake_after_sleep_onset = FloatField()
11    number_of_awakenings = FloatField()
12    average_awakening_length = FloatField()
13    movement_index = FloatField()
14    fragmentation_index = FloatField()
15    sleep_fragmentation_index = FloatField()
16    before_sleep_cortisol = FloatField()
17    before_sleep_melatonin = FloatField()
18    wake_up_cortisol = FloatField()
19    wake_up_melatonin = FloatField()
20    Pittsburgh = IntegerField(null=True)
21
22    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE, related_name="vlasnik")
```

Slika 3.4. Primjer ORM-a u Django

nom view (slika 3.5.). U pogledima se može nalaziti većina logike poslužiteljske strane aplikacije, npr. komunikacija s modelom i bazom podataka, validacija podataka i definiranje metoda. Svaki pogled preslikan je na URL zadan u `urls.py` te može podržati sve metode HTTP-a.

```
class CreateUserView(generics.CreateAPIView):
    permission_classes = [AllowAny]
    def post(self, request, *args, **kwargs):
        serializer_class = UserSerializer(data=request.data)
        if serializer_class.is_valid():
            serializer_class.save()
            return Response(data={'message': 'User registered successfully'}, status=status.HTTP_201_CREATED)
        else:
            return Response(data={'message': 'User registered unsuccessfully'}, status=status.HTTP_400_BAD_REQUEST)
```

Slika 3.5. Primjer pogleda u Django

4. Zaključak

U doba iznimne ubrzanosti života kvalitetan odmor nikada nije bio neophodniji. Iako je san i dalje prilično neobjašnjiv i neistražen, redovito se provode istraživanja o istome. Danas znamo da postoje određeni biomarkeri koji nam mogu nešto lakše predložiti kvalitetu sna. Takvi biomarkeri mogu se koristiti kako bi se osobama moglo pomoći poboljšati kvalitetu sna i time znatno poboljšati kako fizičko tako i mentalno zdravlje. Razvijena aplikacija nudi upravo to. Na temelju istraživanja o snu 22 muškaraca stvorena je baza podataka na temelju koje je napravljen model strojnoga učenja s pomoću kojeg aplikacija za određene ulazne vrijednosti biomarkera daje predikciju za "Pittsburgh sleep quality index", akademski prihvaćenu mjeru kvalitete sna. Model strojnog učenja zasnovan je na tehnici slučajne šume koja za ovako malu količinu i dalje pruža dovoljno dobru predikciju, dok s druge strane ne pati od prenaučenosti. Zaključci koji proizlaze iz rezultata ovog rada pokazuju da je aplikacija koja pomaže ljudima u poboljšanju uvjeta svoga života jednostavna za razvoj, ali da se može i nastaviti razvijati dobivanjem više podataka o fizičkoj aktivnosti osobe s npr. pametnoga sata. Također, rad prikazuje kako se, zbog svoje iznimne dostupnosti, umjetna inteligencija danas može primijeniti na širok spektar problema, pa tako i na onaj koji bi direktno mogao pomoći ljudima.

Literatura

- [1] S. Triantafyllou, S. Saeb, E. G. Lattie, D. C. Mohr, i K. P. Kording, “Relationship between sleep quality and mood: Ecological momentary assessment study”, *JMIR Ment Health*, sv. 6, br. 3, str. e12613, Mar 2019. <https://doi.org/10.2196/12613>
- [2] A. Rossi, E. Da Pozzo, D. Menicagli, C. Tremolanti, C. Priami, A. Sirbu, D. A. Clifton, C. Martini, i D. Morelli, “A public dataset of 24-h multi-levels psycho-physiological responses in young healthy adults”, *Data*, sv. 5, br. 4, 2020. <https://doi.org/10.3390/data5040091>
- [3] A. Tluczek, J. B. Henriques, i R. L. Brown, “Support for the reliability and validity of a six-item state anxiety scale derived from the State-Trait anxiety inventory”, *J. Nurs. Meas.*, sv. 17, br. 1, str. 19–28, 2009.
- [4] C. S. Carver i T. L. White, “Behavioral inhibition, behavioral activation, and affective responses to impending reward and punishment: the bis/bas scales.” *Journal of Personality and Social Psychology*, sv. 67, str. 319–333, 1994. <https://doi.org/10.1037/0022-3514.67.2.319>
- [5] P. J. Brantley, C. D. Waggoner, G. N. Jones, i N. B. Rappaport, “A daily stress inventory: development, reliability, and validity”, *J. Behav. Med.*, sv. 10, br. 1, str. 61–74, veljača 1987.
- [6] V. Tran, *Positive Affect Negative Affect Scale (PANAS)*. New York, NY: Springer New York, 2013., str. 1508–1509. https://doi.org/10.1007/978-1-4419-1005-9_978
- [7] Y. Katsu i M. E. Baker, “Subchapter 123d - cortisol”, u *Handbook of Hormones (Second Edition)*, second edition izd., H. Ando, K. Ukena, i S. Nagata,

- Ur. San Diego: Academic Press, 2021., str. 947–949. <https://doi.org/https://doi.org/10.1016/B978-0-12-820649-2.00261-8>
- [8] D. E. Blask, “Melatonin, sleep disturbance and cancer risk”, *Sleep Medicine Reviews*, sv. 13, br. 4, str. 257–264, 2009. <https://doi.org/https://doi.org/10.1016/j.smr.2008.07.007>
- [9] IBM, <https://www.ibm.com/think/topics/machine-learning>, [mrežno; stranica posjećena: lipanj 2025.].
- [10] O. A. Montesinos López, A. Montesinos López, i J. Crossa, *Overfitting, Model Tuning, and Evaluation of Prediction Performance*. Cham: Springer International Publishing, 2022., str. 109–139. https://doi.org/10.1007/978-3-030-89010-0_4
- [11] scikit learn, <https://scikit-learn.org/stable/>, [mrežno; stranica posjećena: lipanj 2025.].
- [12] Chaya, “(level up coding) random forest regression”, <https://levelup.gitconnected.com/random-forest-regression-209c0f354c84>, [mrežno; stranica posjećena: lipanj 2025.].
- [13] GeeksForGeeks, <https://media.geeksforgeeks.org/wp-content/uploads/20231129130431/11111111.png>, [mrežno; stranica posjećena: lipanj 2025.].
- [14] P. F. Smith, S. Ganesh, i P. Liu, “A comparison of random forest regression and multiple linear regression for prediction in neuroscience”, *Journal of Neuroscience Methods*, sv. 220, br. 1, str. 85–91, 2013. <https://doi.org/https://doi.org/10.1016/j.jneumeth.2013.08.024>

Sažetak

Predviđanje percipirane kvalitete sna temeljem objektivnih mjerenja kvalitete sna metodama strojnog učenja

Borna Zelić

Kvalitetnom analizom i obradom podataka moguće je naučiti model strojnoga učenja da izračuna kvalitetu sna. U ovome radu isprobani su razni modeli strojnoga učenja, a u aplikaciju je ugrađen model slučajne šume, koji je zbog svoje jednostavnosti i otpornosti na prenaučenosť davao najbolje rezultate u vidu najmanjeg korijena kvadrata srednjih vrijednosti pogrešaka. Slobodno dostupni internetski podatci korišćeni za učenje i testiranje modela dobiveni su iz istraživanja kvalitete sna provedenog na 22 muškaraca (<https://physionet.org/content/mrash/1.0.0/>). Algoritam slučajne šume pokazao se precizniji u predikciji u odnosu na algoritme linearne regresije, stabla odluke i Gaussove naivnog Bayes. Interakcija s modelom strojnog učenja postignuta je pomoću mrežnog sućelja razvijenog pomoću radnog okvira Django za poslužiteljsku stranu i radnog okvira React za korisničku stranu. Cijela aplikacija ima cilj približiti zdrav i kvalitetan san svim korisnicima te osvijestiti ih o važnosti dobrog odmora.

Ključne riječi: strojno učenje, slučajna šume, kvaliteta sna, web aplikacija, Django, React

Abstract

Predicting perceived sleep quality based on objective sleep quality measurements using machine learning methods

Borna Zelić

With high-quality data analysis and processing, it is possible to teach a machine learning model to calculate sleep quality. In this paper, various machine learning models were tested, and the application incorporated a random forest model, which, due to its simplicity and resistance to overfitting, gave the best results in terms of the root mean square error values. The freely available online data used for training and testing the model was obtained from a sleep quality study conducted on 22 men (<https://physionet.org/content/mmask/1.0.0/>). The random forest algorithm proved to be more accurate in prediction compared to linear regression, decision tree, and Gaussian naive Bayes algorithms. Interaction with the machine learning model was achieved using a web interface developed using the Django framework for the server side and the React framework for the user side. The entire application aims to bring healthy and quality sleep closer to all users and to make them aware of the importance of a good rest.

Keywords: machine learning, random forest, sleep quality, web application, Django, React