SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA I INFORMATIKA

Vinko Kliček

Diplomski rad

Teorijski izračun i analiza korelacija WiFi signala i broja ljudi pomoću strojnog učenja

Voditelj diplomskog rada: prof. dr. sc. Hrvoje Buljan

Ocjena diplomskog rada: _____

Povjerenstvo:
1. prof. dr. sc. Hrvoje Buljan
2. doc. dr. sc. Vedran Đerek
3. izv. prof. dr. sc. Željko Skoko

Datum polaganja: 28.9.2022.

Zagreb, 2022.

Veliko hvala mentoru prof. dr. sc. Hrvoju Buljanu na ukazanom strpljenju, savjetima, pomoći i razumijevanju tijekom izrade i pisanja ovog diplomskog rada te suvoditeljima Silviju Domazetu i Siniši Nikoliću isto na ukazanoj pomoći.

Hvala mom bratu Filipu i roditeljima na bezuvjetnoj ljubavi i podršci jer niste nikad sumnjali u mene.

Hvala prijateljima i kolegama koji su bili uz mene kad je bilo teško i obilježili moje studentske dane na najljepši način i što ste sudjelovali u eksperimentalnim mjerenjima.

Sažetak

Cilj istraživanja je korištenjem analize WiFi signala otkriti prisustvo ljudi u WiFi polju. Podatci koje analiziramo su intenziteti elektromagnetskog (WiFi) polja u danim točkama prostora, za danu konfiguraciju objekata koji se nalaze u prostoriji. Koristimo tri različite metode strojnog učenja na istim setovima podataka. Gledamo rješenja koja nam daje strojno učenje za pojedinu metodu te ih uspoređujemo s ostalim metodama. Zanima nas koja metoda daje bolje rezultate te što možemo u budućnosti poboljšati da dobijemo bolja rješenja. Jedan set podataka je dobiven teorijski koristeći program COMSOL, dok je drugi set podataka dobiven eksperimentalno.

Ključne riječi:

DTC – Decision Tree Classification

GNB – Gaussian Naive Bayes

EM – elektromagnetski

Theoretical calculation and analysis of WiFi signals with machine learning

Abstract

The goal of the research is to detect the presence of people in the WiFi field using WiFi signal analysis. The data that we analyze are the intensities of the electromagnetic (WiFi) field in given points of space, for a given configuration of objects located in the room. We are using three machine learning methods on same datasets. Machine learning give us results for individual method and then we compare those results with other machine learning results. We do that because we want to see what method works the beast and how we can upgrade to get better results. One dataset was obtained theoretically using the COMSOL program, while the other dataset was obtained experimentally.

Keywords:

DTC – Decision Tree Classification

GNB – Gaussian Naive Bayes

EM – electromagnetic

Sadržaj

1. Uvod

1.1. Definicija problema

U ovom diplomskom analiziramo sljedeći problem. Zamislimo prostoriju u kojoj se nalazi elektromagnetsko (EM) polje. Konkretno ovdje nas zanima WiFi polje odnosno EM zračenje na frekvenciji 2.4 GHz. Kada se ljudi ili drugi objekti nalaze u tom polju, EM valovi se lome i apsorbiraju na ljudima odnosno tim objektima. Refleksija, lom odnosno apsorpcija utječu na intenzitet u prostoru od interesa. Budući da u realističnim primjenama ne možemo mjeriti intenzitet u svim točkama prostora već samo na onim lokacijama gdje se nalaze detektori EM zračenja, nije jednostavno rekonstruirati zbog toga što se zbiva u EM polju, odnosno zašto je došlo do promjene intenziteta. U tom kontekstu nam pomažu procesi strojnog učenja. Naime, plan je naučiti stroj da iz intenziteta na mjestima gdje se nalaze detektori prepozna koliko se ljudi nalazi u WiFi polju. Ovaj je cilj preambiciozan za jedan diplomski rad, no usmjeriti ćemo istraživanja prema tom cilju. Samo strojno učenje ćemo pisati u programskom jeziku Python.

1.2. Python

Python spada u više programske jezike [10]. Lako ga je naučiti i koristiti, u njemu možemo programirati u različitim stilovima poput: objektno orijentiranog programiranja, strukturno i aspektno orijentirano programiranje itd. Python podražava sve vrste operacijskih sustava što ga čini dobrim za „cross platform“ programiranje. Python koristi interpreter što znači da kod izvršava liniju po liniju za razliku od kompajlera gdje se cijeli kod izvršava odjednom.

1.3. Strojno učenje

Strojno učenje je vrsta umjetne inteligencije gdje računalo uči na temelju dobivenih podataka predviđa rješenje. Postoje četiri vrste strojnog učenja: nadzirano strojno učenje, nenadzirano strojno učenje, polu nadzirano strojno učenje i „Reinforcement“ strojno učenje [11]. Strojnom učenju u većini slučajeva je poželjno da baza podataka na kojem ući bude čim veća i raznovrsnija jer s time dobivamo veću preciznost. Ovdje bazu podataka čine: a) teorijski računi dobiveni u COMSOLU, i b) stvarna mjerenja.

1.3.1. Nadzirano strojno učenje

U nadziranom strojnom učenju algoritam uči iz input/output parova. Čovjek priprema te parove to jest vrši normalizaciju podataka. Normalizacija podataka je brisanje istih input/output parova, sparivanje podataka u input/output parove i brisanje podataka koji nemaju para. To može biti jako zahtjevno, ali je zato algoritam efikasan. Osnovna ideja nadziranog strojnog učenja je modeliranje procesa pomoću matematičkih funkcija. Svaka takva funkcija ovisi o određenom broju parametara. Učenje je proces promjene tih parametara pri čemu se mijenja oblik funkcije tako da ona sve bolje predstavlja podatke s kojima radimo. Kod nadziranog strojnog učenja postoji puno gotovih modela algoritama. Ovdje ćemo navesti i koristiti neke od njih.

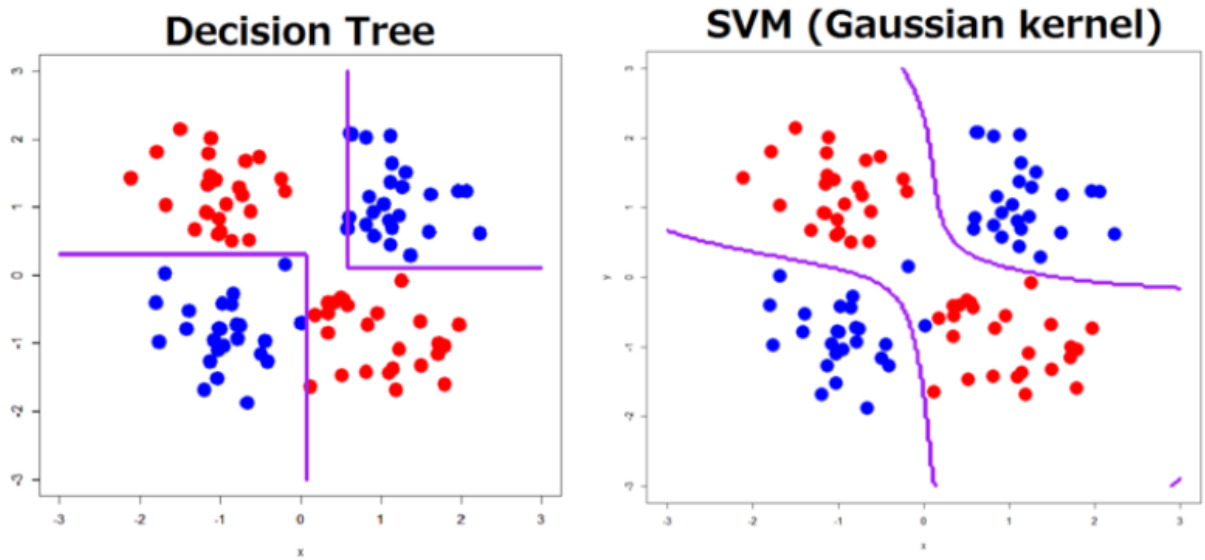
1.3.1.1. Decision Trees

Decision Trees je neparametarski nadzirano strojno učenje koje se koristi za klasifikaciju i regresiju. Svrha Decision Treesa je stvoriti model koji predviđa rezultat na temelju ulaznih varijabli učenjem jednostavnih pravila odlučivanja koje zaključuje iz podataka. Stablo možemo gledati kao diskretnu aproksimaciju skupa zadanih točaka [1].

Prednosti Decision Treesa je da su lagana za razumjeti i objasniti, odnosno stabla možemo vizualizirati. Nadalje, ova metoda zahtjeva malu pripremu podataka, dok ostali modeli najčešće zahtijevaju normalizaciju podataka. Cijena korištenja stabla je logaritamska u broju podataka korištenih za učenje stabla. Može koristiti numeričke i kategorične podatke. Treba napomenuti da scikit-learn implementacija za sad ne podržava kategorične podatke. Predviđa više od jednog rješenja istovremeno i koristi white box model[1]. White-Box je model čija su unutarnja logika, rad i programski koraci transparentni i stoga je proces donošenja odluka interpretabilan. [9]. Vraća nam statističke podatke što nam omogućuje da vidimo kolika nam je pouzdanost modela. Daje nam dobre rezultate iako nismo napravili dobru normalizaciju podataka..

Mane Decision Treesa je među ostalim da napravi prekompleksno stablo koje dobro ne generalizira podatke. Kada dođe do takve situacije to se zove overfitting. Sa mehanizmima poput obrezivanja, zadavanje minimalnog broja uzoraka potrebnog na čvorištu ili postavljanjem maksimalne dubine stabla su nam potrebna da bi overfitting izbjegli. Stabla mogu biti nestabilna

zbog male varijacije u podacima što uzrokuje da se potpuno različita stabla generiraju. Taj problem ublažavamo tako da stabla koristimo u ansamblu. Predviđanja stabla nisu glatka niti su kontinuirana te zato nisu doba za ekstrapolaciju. Koncepti poput XOR, paritet ili multiplexer stablo teško uči zato što ih nije lagano izraziti[1]. Uzmimo npr. XOR. Ako XOR ima dvije ulazne varijable dobiti ćemo vrlo jednostavno stablo istinitosti, ali kada povećamo broj ulaznih varijabli stablo postaje jako kompleksno i duboko što nije optimalno. To se dogodi jer stabla, kao što smo prije spomenuli, nisu glatka. Posljedica toga je puno rubnih slučajeva zbog XOR distribucije, a mi želimo imati takvu granicu koja će to izbjeći. Slika 1 prikazuje naš primjer[12].



Slika 1 XOR set podataka za klasifikaciju[12].

1.3.1.2. Naive Bayes

Naive Bayes algoritam baziran je na primjeni Bayesovog teorema sa „naivnom“ pretpostavkom da svaki par podataka je nezavisan. Bayesov teorem glasi: imamo potpun sustav događaja $x_1, \dots, x_n$, a y je realizirani događaj:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}$$

Koristimo pretpostavku da je svaki pat podataka nezavisan zaključujemo:

$$P(x_i \mid y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y)$$

Za svaki i dobijemo sljedeći izraz:

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod \quad_{i=1}^{n} P(x_i \,|y)}{P(x_1, \dots, x_n)}$$

$P(x_1, \dots, x_n)$ nam je uvijek isti jer se ulazni podaci ne mijenjaju pa možemo koristiti pravilo klasifikacije:

$$P(y \mid x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

$$\hat{y} = arc \max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

Sada možemo iskoristiti Maximum A Posteriori pretpostavku da pretpostavimo $P(y)$ i $P(x_i|y)$ [2].

Razlika između naive Bayes klasifikatora je u pretpostavki kakva će biti distribucija $P(x_i \,|y)$. Iako je model jako pojednostavljen naive Bayes radi vrlo dobro. Potrebna mu je mala količina podataka za učenje. Naive Bayes je puno brži u usporedbi sa više sofisticiranim metodama[2].

1.3.1.3. Linearni Model

U linearnom modelu svi modeli su napravljeni za regresiju za koju je rješenje linearna kombinacija ulaznih podataka. U matematičkoj notaciji, ako je $\hat{y}$ predviđena vrijednost:

$$\hat{y}(\omega, x) = \omega_0 + \omega_1 x_1 + \cdots + \omega_p x_p$$

Odredimo vektore $\omega = (\omega_1, \dots, \omega_n)$ kao koeficijente i ω_0 kao sjecište y-osi. $\hat{Y}$ je ovisna varijabla, x_n je neovisna varijabla. Uočavamo da se rješenje $\hat{y}$ mijenja o ovisnosti o varijablama x_n [3].

1.4. Scikit-learn

Scikit-learn ili skraćeno Sklearn je robusna biblioteka u Python programskom jeziku za strojno učenje. Ta biblioteka nam pruža spektar efikasnih alata za strojno učenje i statističko modeliranje uključujući klasifikaciju, regresiju, klasteriranje i dimenzionalnu redukciju. Biblioteka je većinom napisana u Pythonu, napravljena je na *NumPy*, *SciPy* i *Matplotlipu*.

Da bismo mogli koristiti ovu biblioteku moramo zadovoljiti sljedeće uvjete:

- Python 3.5 ili noviji
- NumPy 1.11.0 ili noviji

- SciPy 0.17.0 ili noviji
- Joblib 0.11 ili noviji
- Matplotlib 1.5.1 ili noviji, potreban nam je za crtanje u Sklearnu
- Pandas 0.18.0 ili novije, za manipulaciju podataka.

Da bismo koristili Sklearn biblioteku prethodnu ju je potrebno instalirati jer ne dolazi u osnovnom Python paketu [4].

2. Cilj istraživanja

Cilj istraživanja je pomoću strojnog učenja odrediti koliko se ljudi nalazi u WiFi polju iz intenziteta na mjestima gdje se nalaze detektori elektromagnetskog zračenja u nekom prostoru. Konkretno pomoću metoda Decision Tree Classification, Gaussian Naive Bayes i linearne regresije. Korelacija dobivenih i očekivanih rezultata s tim metodama. Utjecaj broja ljudi na intenzitet EM zračenja na frekvenciji 2.4 GHz te njihovo ponašanje u prostoru, je li se kreću po prostoru ili miruju ili kombinacija tih dviju mogućnosti. Mjerenja su vršena u numeričkoj simulaciji COMSOL i eksperimentalno.

3. Metode

3.1. Decision Tree Classification

Decision Tree Classification je metoda u Sklearn biblioteci. Ova metoda može raditi multi-class klasifikaciju što znači da predviđa više klasa. Kao i u ostalim klasifikatorima DTC uzima dvije liste, jednu listu X koja je rijetka ili gusta u obliku (n_samples, n_features), sadrži podatke na kojima uči. Druga lista Y u obliku (n_samples), sadrži rješenja na kojima uč[1,5]i.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

model = DecisionTreeClassifier()
model.fit(X_train, y_train)

predictions = model.predict(X_test)

accuracy score(predictions,y test)
```

Kod 1 Decision Tree Classification

Funkcija *train_test_split* nam služi da napravio set podataka za učenje i set podataka za testiranje. Funkcija nasumično uzima podatke na kojima će učiti ili testirati. Pomoću parametra *test_size* određujemo postotak od ukupnog broja podataka za testiranje[6]. U našem slučaju 20% podataka koristimo za testiranje, a ostatak je za učenje. U sljedećem redu inicijaliziramo metodu. Sa funkcijom *fit* učimo model. U varijablu *predictions* pospremamo predviđanja. Funkcija *predict* radi predviđanja na temelju podataka iz seta *X_test*. Funkcija *accuracy_score* uspoređuje predviđene podatke sa podacima koji su trebali biti te nam vraća preciznost u rasponu od 0.0 do 1.0. Funkcija je namijenjena za klasifikatore [7].

3.2. Gaussian Naive Bayes

Kod Gaussian Naive Bayes koristimo sljedeću vjerojatnost:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\sigma_y^2}\right)$$

gdje se za parametre μ_y i σ_y pretpostavlja da imaju najveću vjerojatnost [2].

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import GaussianNB

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

gnb = GaussianNB()
gnb.fit(X_train, y_train)

predictions = gnb.predict(X_test)

accuracy score(predictions,y test)
```

Kod 2 Gaussian Naive Bayes

Funkcija *train_test_split* radi isto kao u prethodnoj metodi [Decision Tree Classification]. Ovdje isto koristimo 20% podataka za testiranje, a ostatak za učenje. Inicijaliziramo metodu, sa funkcijom *fit* učimo model. U varijablu *predictions* pospremamo predviđanja iz seta podataka u *X_test*. Funkcijom *accuracy_score* nam vraća preciznost modela [7].

3.3. Ordinary Least Squares

Metodom *LinearRegression* fita linearni model sa koeficijentima $\omega = (\omega_1, \dots \omega_p)$ da minimizirao rezidualnu sumu kvadrata između seta podataka za učenje i seta podataka za

testiranje koji su predviđeni linearnom aproksimacijom. Matematički rješavamo sljedeći problem:

$$\min_{\omega} ||X\omega - y||_2^2$$

LinearRegression će za funkciju *fit* uzeti liste X i y te će koeficijentima ω linearnog modela pospremiti u *coef_* član [3].

```
from sklearn import  linear_model
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split


X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)


regr = linear_model.LinearRegression()
regr.fit(X_train, y_train)


predictions = regr.predict(X_test)


r2 score(predictions,y test)
```

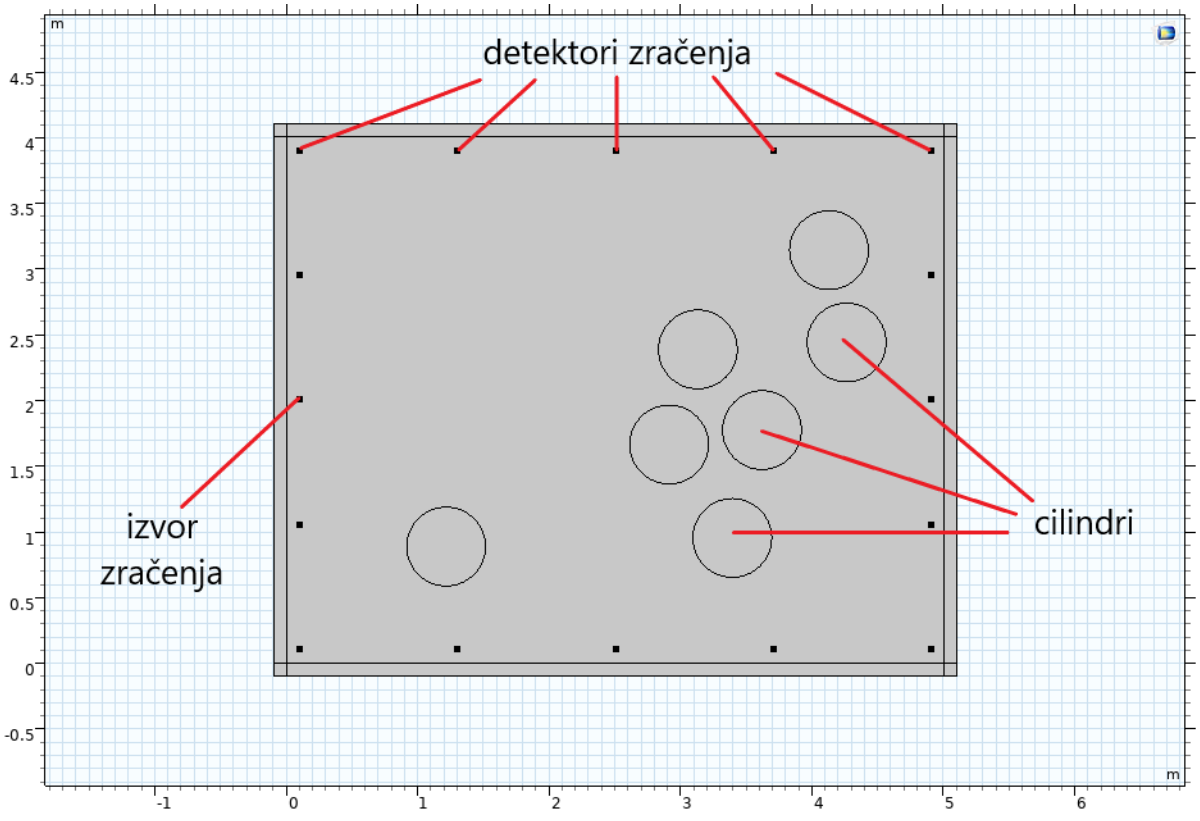
Kod 3 Ordinary Least Squares

Dijelimo setove podataka kao i u prethodnim metodama pomoću funkcije *train_test_split*. Inicijaliziramo metodu. *Fit* funkcija koristimo za učenje modela. U varijablu *predictions* pospremamo predviđanja iz seta podataka u *X_test*. Funkcija *r2_score* nam mjeri preciznost modela. Ovu funkciju koristimo zato što radimo regresiju [8]. Da koristimo funkciju iz prethodnih modela dobili bi grešku jer nisu namijenjeni za regresiju. Funkcija *r2_score* vraća vrijednost od 0.0 do 1.0.

4. Mjerenja i numeričke simulacije

4.1. Numeričke simulacije

U prostoriji se nalazi izvor EM zračenja to jest WiFi signala na frekvenciji 2,4 GHz. Sustav je modeliran korištenjem numeričkog programa u COMSOLU. Prostorija je pravokutna sa betonskim zidovima u dvije dimenzije. Cilindri u prostoriji predstavljaju ljude. Cilindri su ispunjeni vodom i radijusa su 30 cm (to ima smisla jer su ljudi uglavnom načinjeni od vode, a cilindar je aproksimacija z oblik čovjeka). Treća dimenzija nam je beskonačna kako bi račun bio jednostavniji. U prostoriji imamo 15 detektora i jedan izvor. Detektori su postavljeni sa međusobnim razmacima koji su jednaki. Na Slici 2. prikazana je skica prostorije te su naznačene točke u kojima su detektori i izvor. Izvor je beskonačna žica u z smjeru koja daje dipolno zračenje. Detektori bilježe vrijednost intenziteta EM polja.



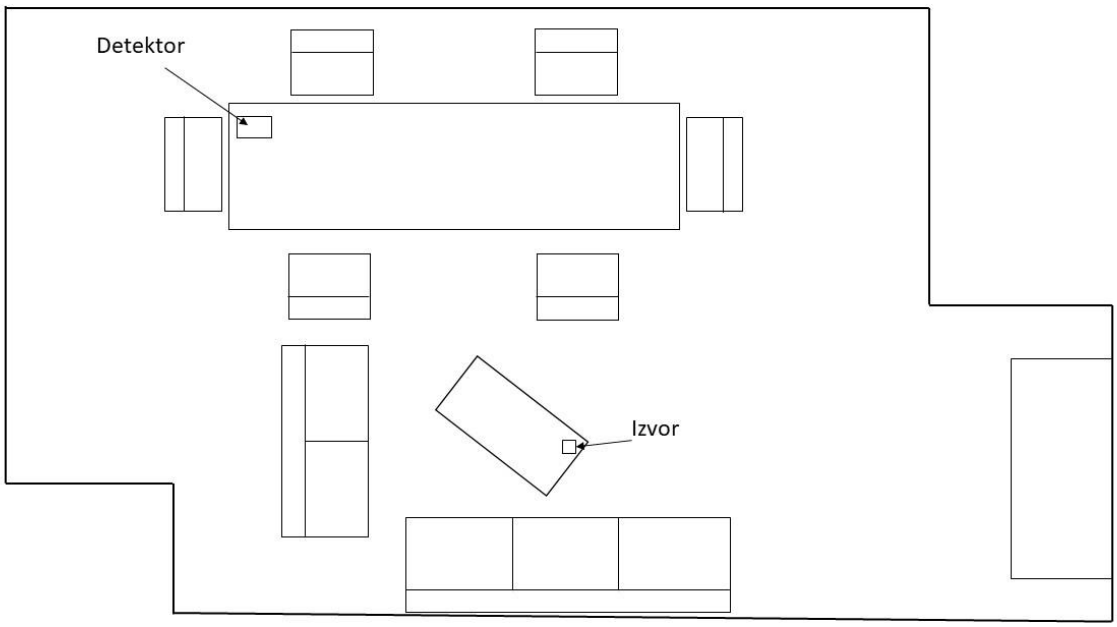
Slika 2 Skica prostora, razmještaj izvora, detektora i cilindara

Zbog ljudske prisutnosti (zbog cilindara) dolazi do raspršenja EM zračenja koje emitira izvor, te je moguća refleksija koja pojačava signal u dijelu prostora ili atenuacija pri čemu signal slabi u dijelu prostora. Razmještaj za svaku simulaciju je nasumičan. Za svaki broj cilindara je

rađen veći broj simulacija u setovima po deset različitih konfiguracija i tako četiri puta za svaki broj ljudi. Rađene su simulacije za 1, 3, 5, 7 i 9 ljudi odnosno cilindara.

4.2. Mjerenja s pravim ljudima

U prostoriji se nalazi izvor EM zračenja to jest WiFi signala na frekvenciji 2,4 GHz. Prostorija je nepravilnog oblika sa ciglenim zidovima. U mjerenju gledamo dvije dimenzije. Treću dimenziju visinu zanemarujemo. Korišteni su pravi ljudi. U prostoriji je jedan detektor i jedan izvor. Na Slici 3 prikazana je skica prostorije te su naznačene točke u kojima su detektori i izvor. Izvor je komercijalni IEEE 802.11 uređaj. Detektor bilježi vrijednost EM polja.



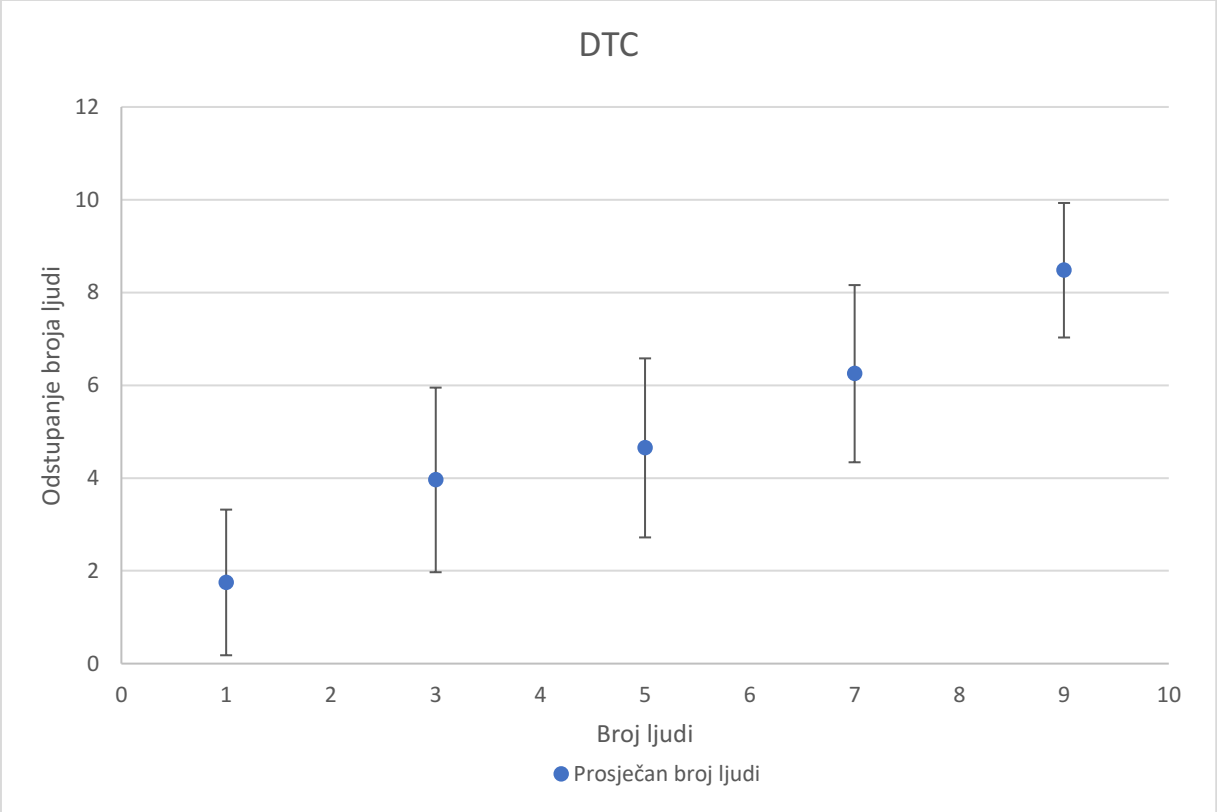
Slika 3 Skica prostora, razmještaj izvora, detektora i ljudi

Kao i za mjerenje u simulatoru Zbog ljudske prisutnosti dolazi do raspršenja EM zračenja koje emitira izvor, te je moguća refleksija koja pojačava signal u dijelu prostora ili atenuacija pri čemu signal slabi u dijelu prostora. Razmještaj ljudi u prostoru je bio nasumičan. Za svaki broj ljudi rađena su tri seta podataka od kojih dva svi miruju, a za jedan kreću se po prostoriji. Set sadrži 1120 mjerenja. Rađena su mjerenja za 1, 2, 3, 4, 5, 6, 7 i 8 ljudi. Iznimka je set za jednu osobu tamo su samo dva seta jedan gdje miruje, a drugi gdje se osoba kreće.

5. Rezultati

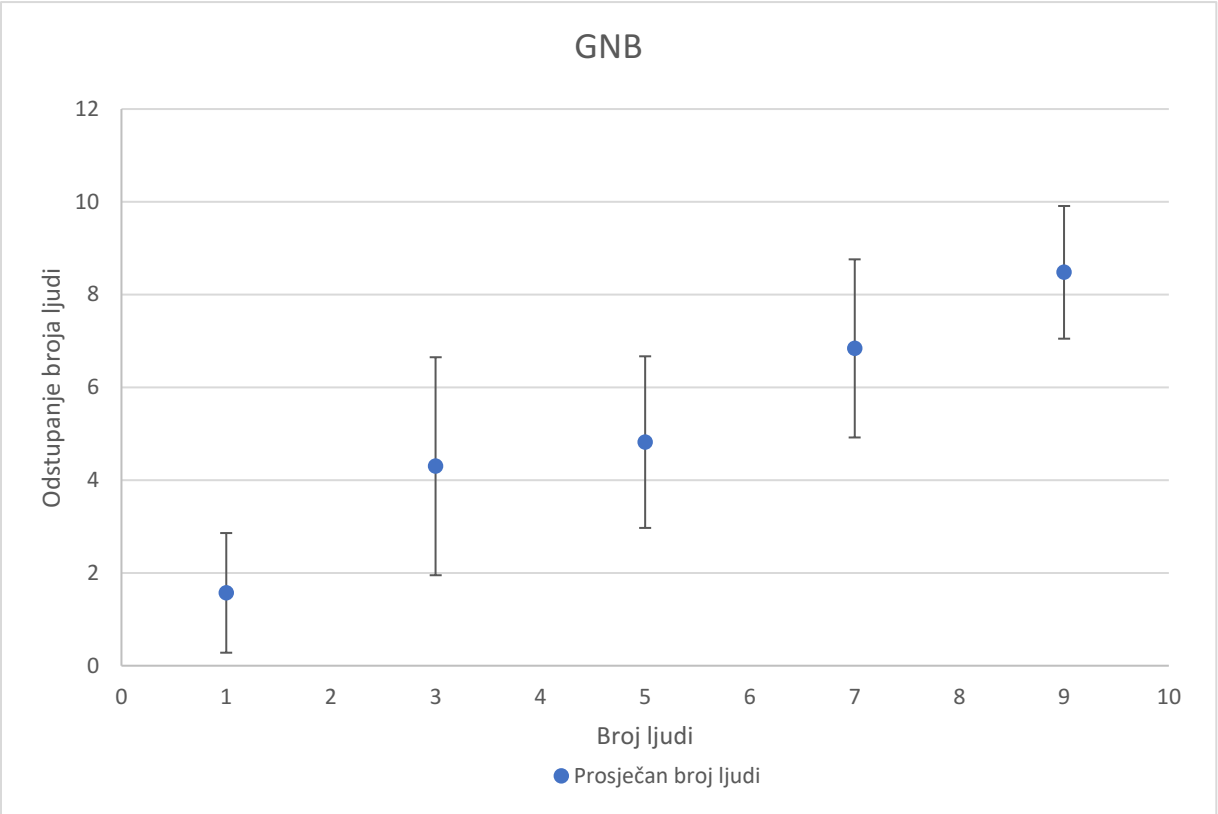
5.1. Rezultati numeričke simulacije

Za svaku metodu sam pokrenuo program sto puta. Pomoću toga sam odredio srednju vrijednost i odstupanje.



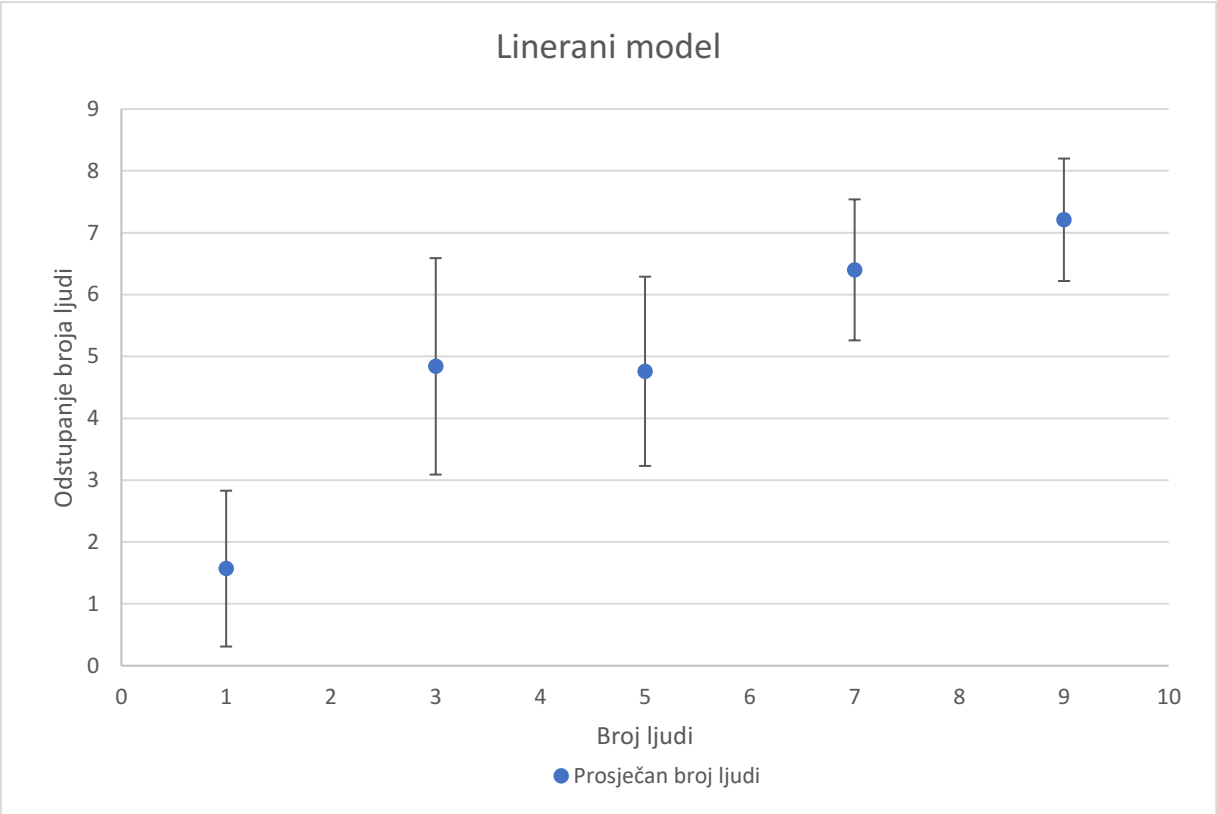
Graf 1 Pogreška strojnog učenja po ljudima u DTC-u

Graf 1 prikazuje predikciju broja ljudi dobivenih DTC metodom za 100 simulacija u odnosu na stvaran broj ljudi koji je prikazan na x osi. Iz Grafa 1 vidimo da je prosječna vrijednost za 100 simulacija vrlo blizu pravoj vrijednosti broja ljudi u prostoriji. Metoda griješi praktički za sve slučajeve $\pm$ 2 osobe što je donekle zadovoljavajuće.



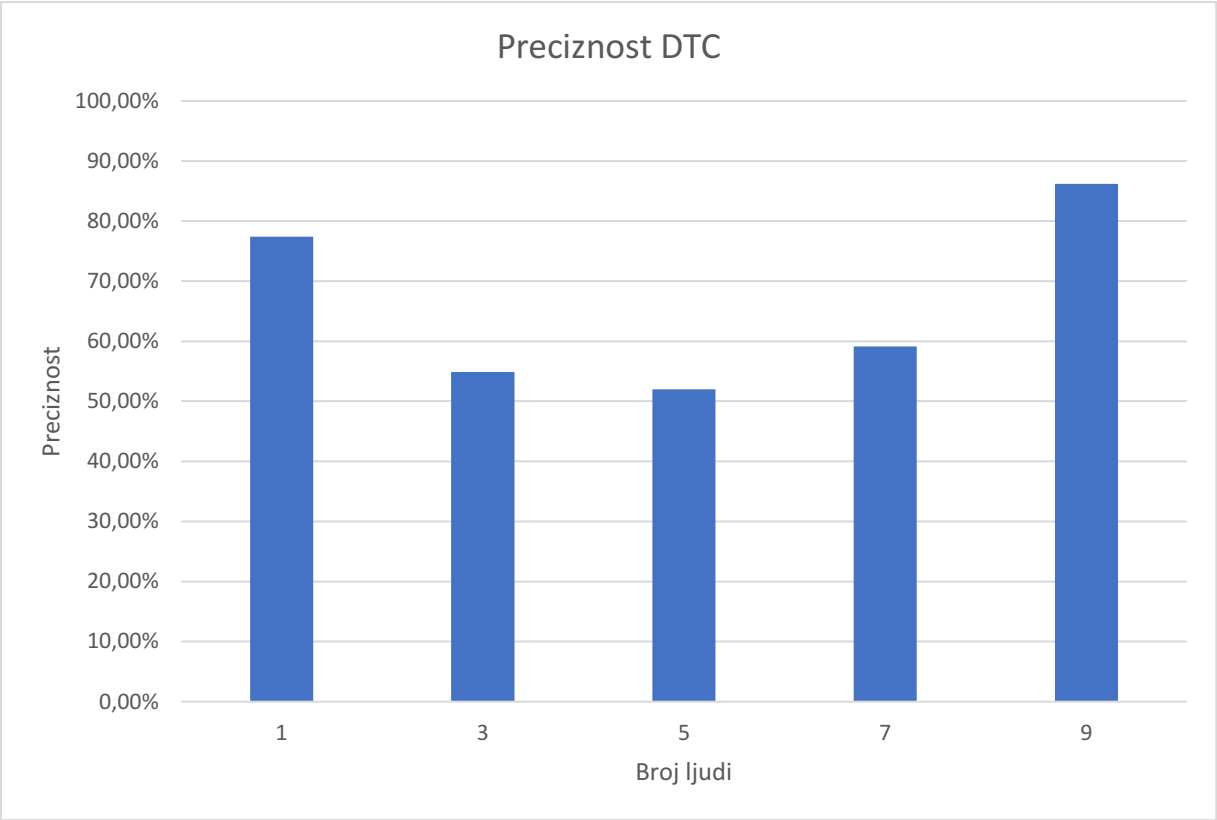
Graf 2 Pogreška strojnog učenja po ljudima u GNB-u

Ovdje prikazujemo predikciju za 100 simulacija dobivenih Gaussian Naive Bayes metodom, u odnosu na stvaran broj ljudi. Ova nam metoda također daje dobre srednje vrijednosti i odstupanja od prave vrijednosti broja ljudi u prostoriji. Model griješi ± 2 osobe. To nam je i dalje donekle zadovoljavajuće.



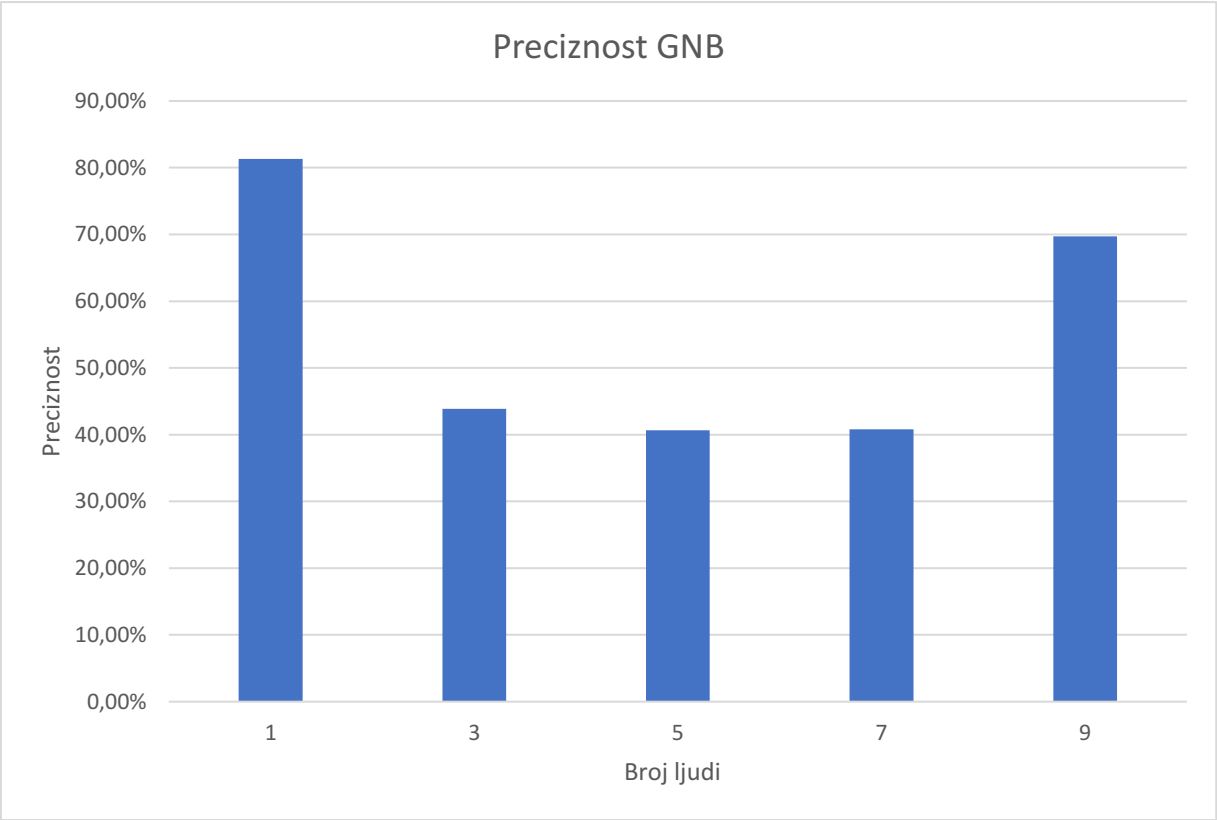
Graf 3 Pogreška strojnog učenja po ljudima u linearnoj regresiji

Kod linearne regresije (Graf 3) rezultati nisu dobri zato što metoda želi učiti iz kontinuiranih vrijednosti dok mi kao ulaz dajemo diskretne vrijednosti. Srednje vrijednosti ljudi imaju velika odstupanja od pravog broja ljudi. Tako da nam ni odstupanje broja ljudi ništa puno ne govori.



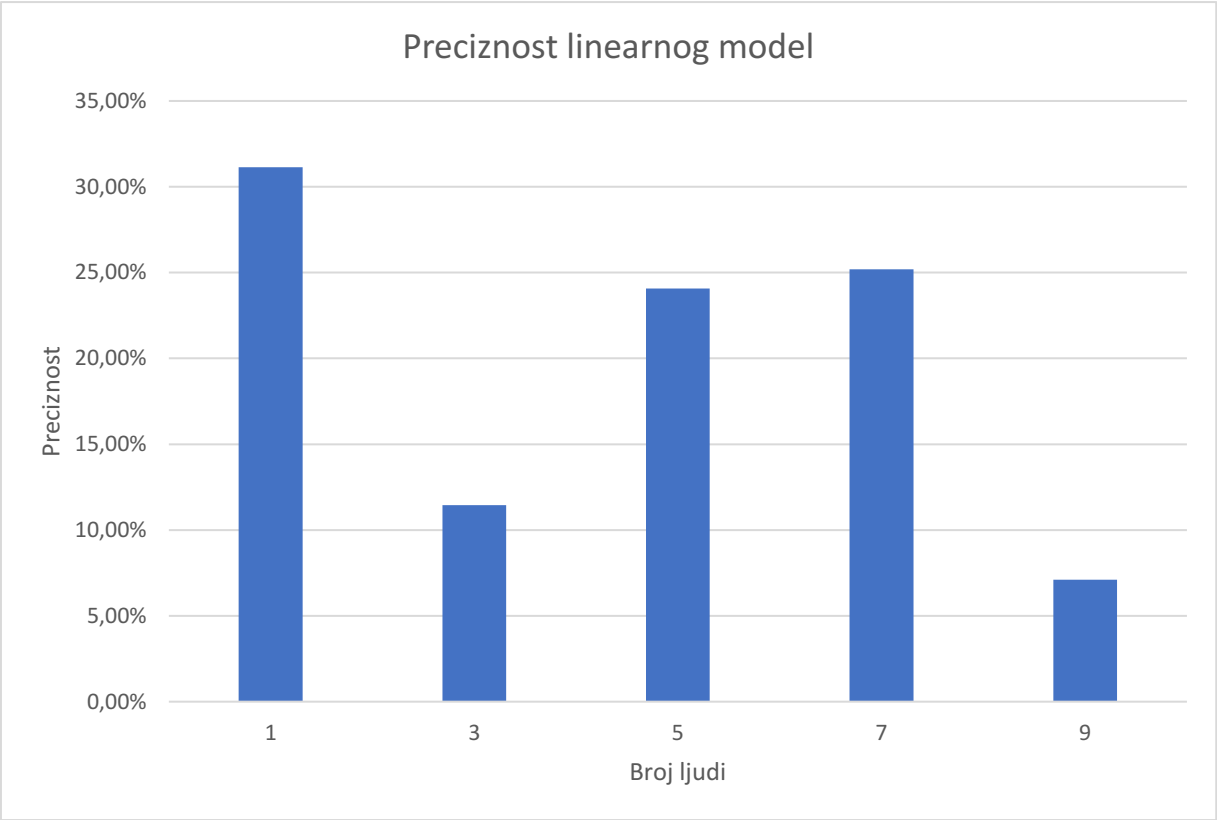
Graf 4 Preciznost DTC po broju ljudi

Iz Grafa 4 možemo zaključiti da preciznost za jednu i devet osoba je velika dok je za ostale slučajeve manja, ali dalje zadovoljavajuća. Vjerojatno se ovdje radi o tome da je kod rubnih situacija pogreška moguća samo u jednom smjeru. Model nam jako ovisi u ulaznim podacima pa ako nam se vrijednosti detektora preklapaju modelu je tada teško napraviti dobru predikciju. Ukupna preciznost modela je $65{,}92\% \pm 7{,}54\%$.



Graf 5 Preciznost GNB-a po broju ljudi

Graf 5 je u strukturi identičan Grafu 4, no za GNB metodu. Preciznost za jednu i devet osoba je vrlo dobra, ali za tri, pet i sedam osoba preciznost nije zadovoljavajuća. Preklapanje vrijednosti detektora ovdje još više utječu na preciznost. To nas ne iznenađuje jer model radi Gaussovu krivulju za svaki broj ljudi. Te ako nam se preklapaju vrijednosti za pojedini broj ljudi nećemo dobiti lijepu Gaussovu krivulju, a ni time dobra predviđanja. Ukupna preciznost modela je $55{,}4\% \pm 6{,}9\%$.

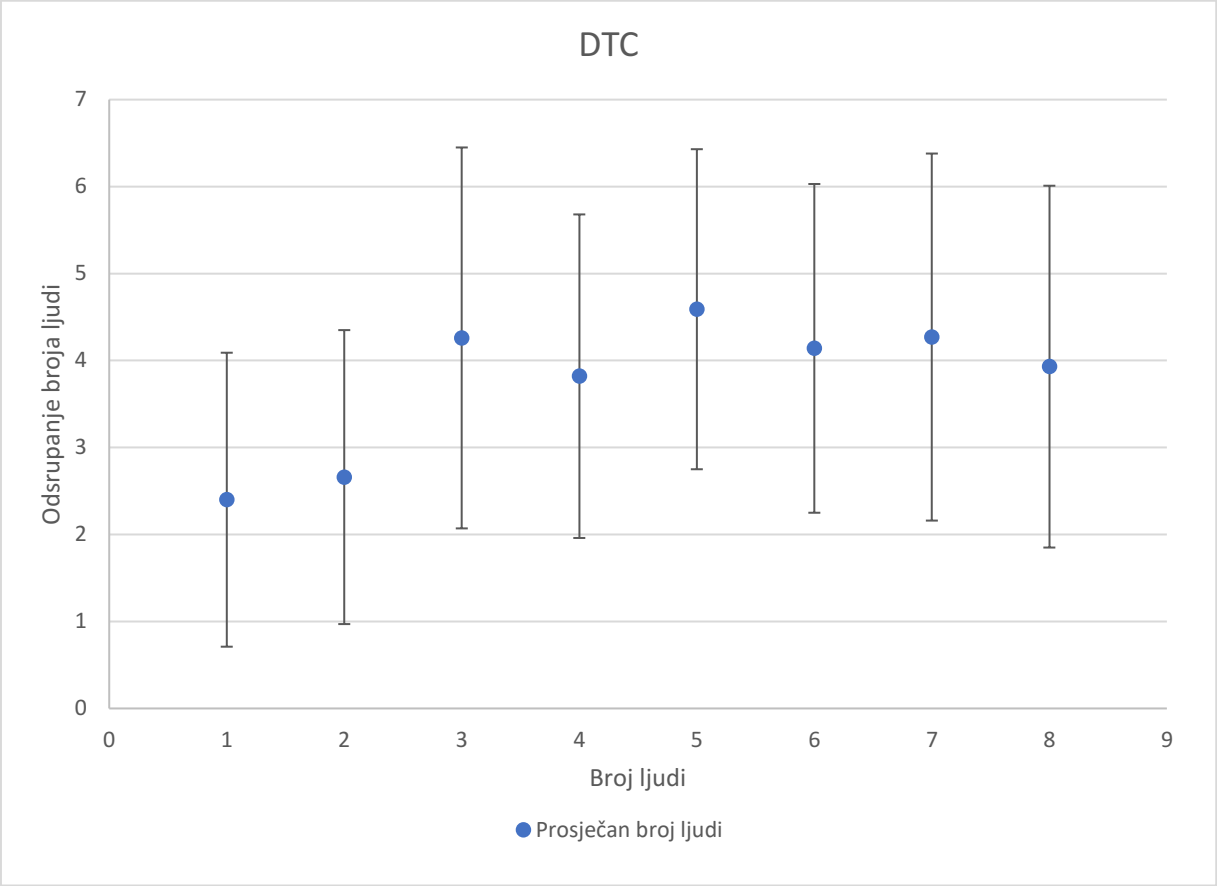


Graf 6 Preciznost linearne regresije po broju ljudi

Na Grafu 6 vidimo da je linearni model jako neprecizan. Gledajući Graf 3 ovakva preciznost je bila očekivana. Možemo zaključiti da linearna regresija nije dobra metoda za rješavanje ovakve vrste problema. Ukupna preciznost modela je $37{,}37\% \pm 16{,}75\%$.

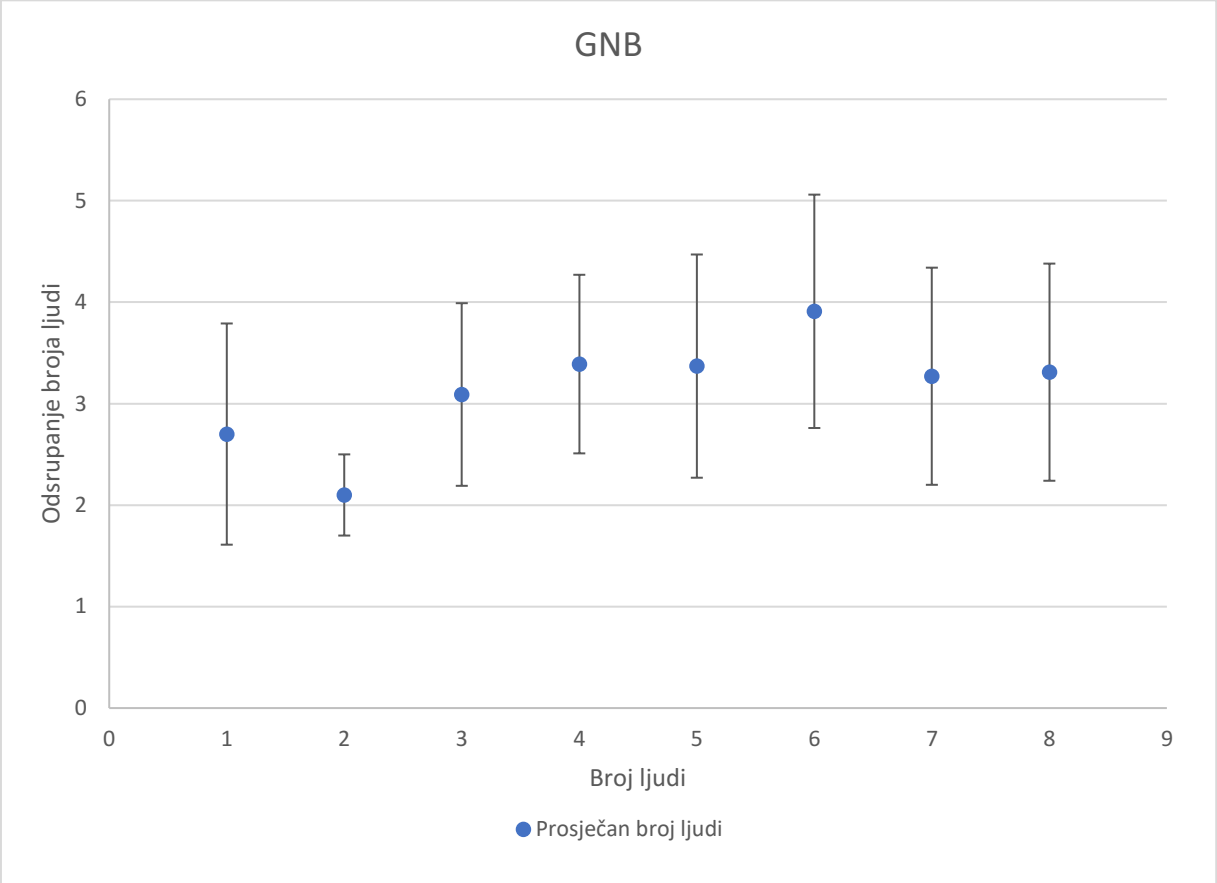
5.2. Rezultati s ljudima

Za svaki set podataka uzeli smo po deset vrijednosti i napravili srednju vrijednost. Tako da je konačan broj podataka po setu 112.



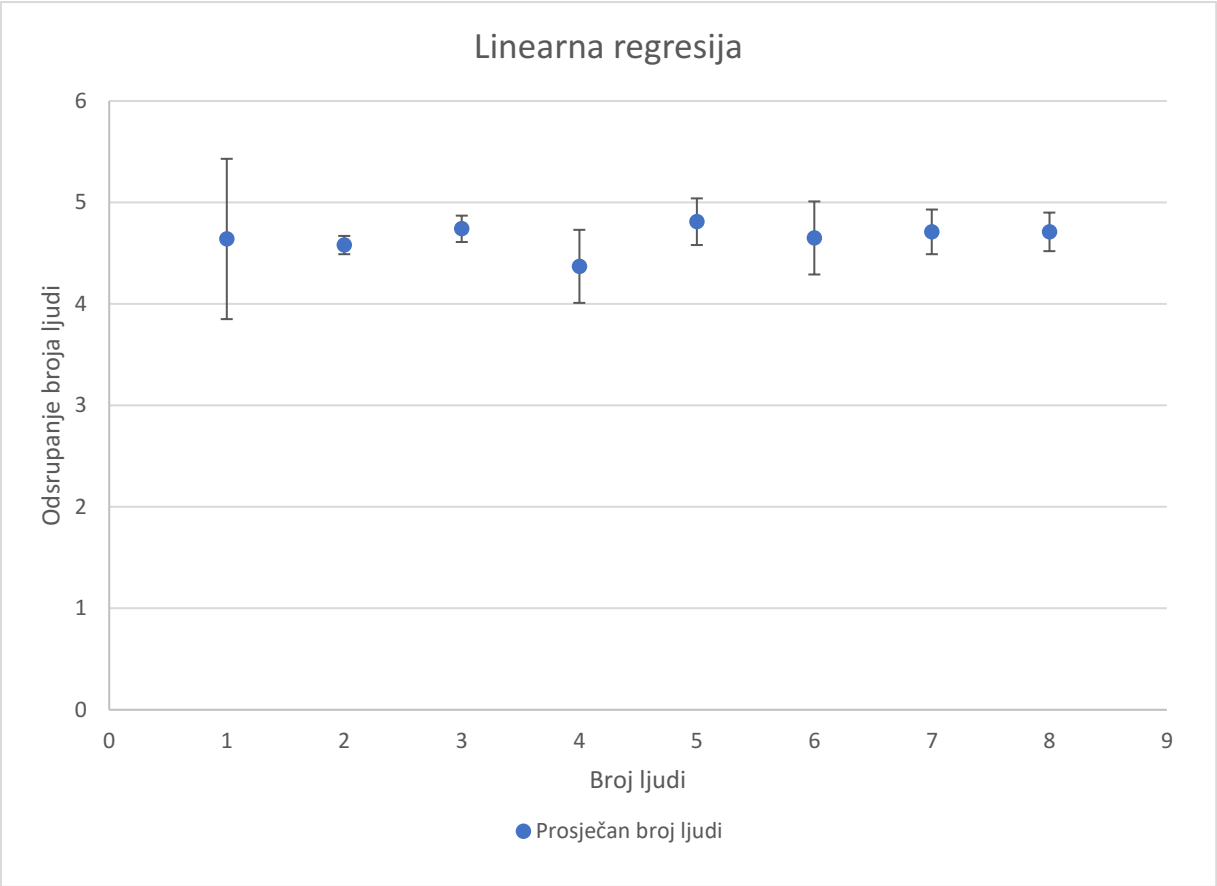
Graf 7 Pogreška strojnog učenja po ljudima u DTC

Graf 7 nam prikazuje velika odstupanja srednje vrijednosti od prave vrijednosti što znači da model na radi dobro. Model ne radi dobro zato što se puno podataka preklapa te je onda modelu teško razlučiti što je pravo rješenje. Nadalje, kod pravih mjerenja imali smo na raspolaganju nažalost samo jedan detektor, pa je strojno učenje učilo iz samo jednog detektora. Za očekivati je da bi realistična mjerenja sa 15 detektora kao ona koja smo imali u numeričkim simulacijama, davala puno kvalitetniji input za strojno učenje, pa bi očekivano dobili bolje rezultate.



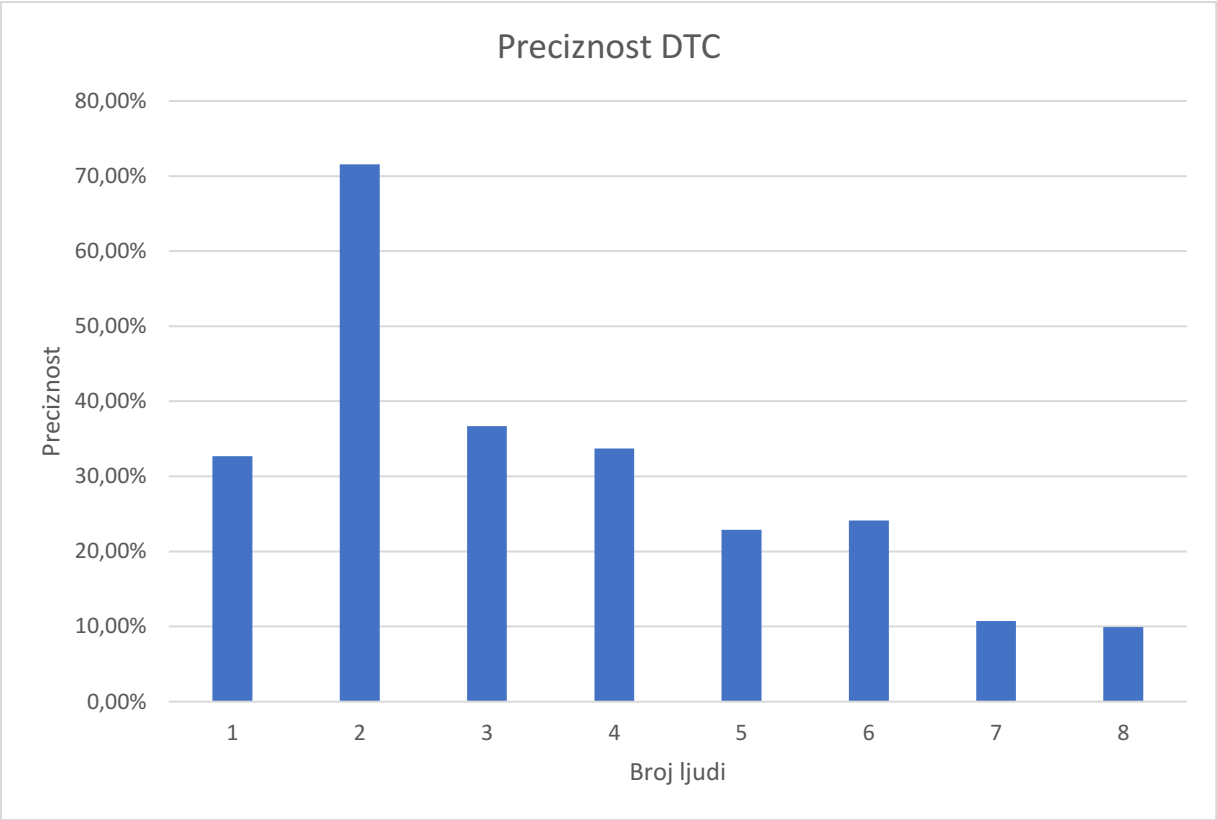
Graf 8 Pogreška strojnog učenja po ljudima u GNB

Graf 8 je jednako loš kao i Graf 7 no ovdje za GNB metodu. U prijašnjem poglavlju [5.1] smo vidjeli da je GNB više osjetljiv na preklapanje podataka. Taj isti razlog uzrokuje ovako loše rezultate.



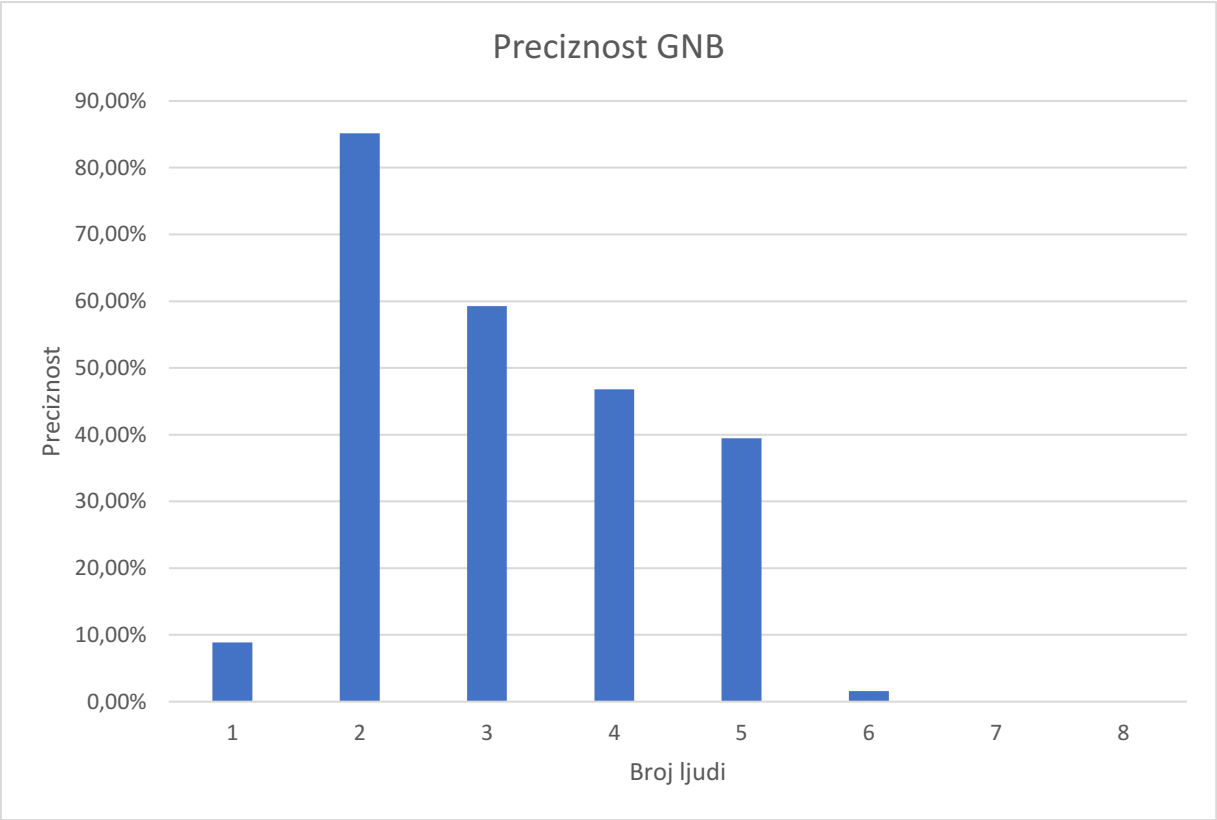
Graf 9 Pogreška strojnog učenja po ljudima u DTC

Ponovno je linearna regresija daje najlošija rješenja kao što vidimo u Grafu 9. Zaključujemo iz grafa da za bilo koju vrijednost detektora dobivamo nazad rješenje da su u prostoriji 4 osobe. Zbog toga odstupanje od srednje vrijednosti je malo.



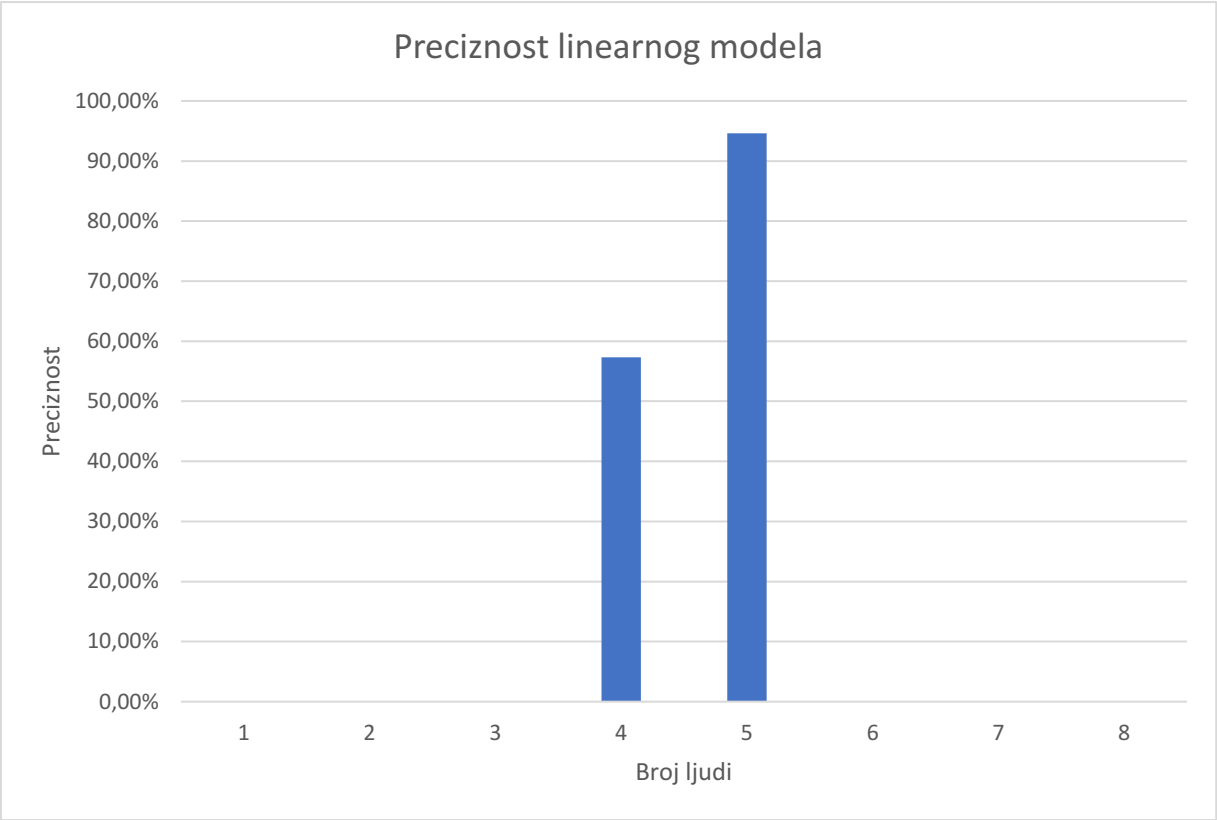
Graf 10 Preciznost DTC-a po broju ljudi

Preciznost prikazana na Grafu 10 je jako loša. Iznimka je preciznost modela za dvije osobe, ali to je nedovoljno da možemo reći da model dobro radi. Ponovno zbog preklapanja podataka dobivamo ovakvu lošu preciznost. Gledajući Graf 9 rezultati su očekivani. Ukupna preciznost modela je $30{,}59\% \pm 1{,}59\%$.



Graf 11 Preciznost GNB-a po broju ljudi

Kao što vidimo na Grafu 11 preciznost je veća za dvoje, troje, četvero i petero ljudi u odnosu na Graf 10, ali zato je lošija za sve ostale varijante. Usporedivši ovaj Graf 11 sa Grafom 8 vidimo da niti jedna predikcija modela nije bila dobra za sedam i osam osoba. Ukupna preciznost modela je $30.89\% \pm 1{,}76\%$.



Graf 12 Preciznost linearne regresije po broju ljudi

Graf 12 je takav zato što su sva predviđanja za četiri ili pet osoba. Što znači da će model predvidjeti četiri ili pet osoba u prostoriji makar ih je u prostoriji dvoje. Model je izrazito loš. Kao i kod mjerenja u simulatoru ovaj model nije dobar za rješavanje ovakve vrste problema. Ukupna preciznost modela je $-4195{,}87\% \pm 1512{,}43\%$. Iz ukupne preciznosti vidimo da je model beskoristan i loš.

6. Zaključak

Iz ovog istraživanja možemo zaključiti da strojno učenje u kombinaciji sa mjerenjima intenziteta WiFi signala ima potencijala za brojanje ljudi u prostoru WiFi polja. Nadzirano strojno učenje jako ovisi u ulaznim podacima. Ovdje smo ulazne podatke dobili na 2 načina: (a) iz numeričkih simulacija i (b) iz eksperimentalnih mjerenja. Rezultati iz simulacija su bili puno bogatiji u smislu da smo ih dobili iz 15 „detektora“, dok su stvarna mjerenja učinjena samo sa jednim detektorom. Stoga su metode strojnog učenja očekivano bolje radile na numerički dobivenim ulaznim podatcima. Preciznije, što se tiče eksperimentalnih podataka, usprkos provedenoj normalizaciji ulaznih podataka dobivamo loša rješenja. Iz toga možemo zaključiti da moramo nabaviti veći broj detektora za preciznije rezultate. Jedan detektor nije dobro koristiti jer ako na primjer zaklonimo detektor dobivamo rezultate kao da je u prostoriji jako puno ljudi, a ubiti mogu u prostoriji biti dvije ili tri osobe. To čini mjerenja jako nepreciznima.

Setove podataka numeričke simulacije normaliziramo te ih simuliramo 100 puta metodama DTC, GNB i linearnom regresijom. Iz tih 100 simulacija dobivamo preciznost po broju ljudi, ukupnu preciznost metode, srednju vrijednost ljudi te njihovo odstupanje od srednje vrijednosti to jest za koliko ljudi metoda pogriješi za određeni broj ljudi. To isto smo napravili i za eksperimentalna mjerenja.

DTC metoda se je pokaza donekle zadovoljavajućom jer ima dobru ukupnu preciznost, dobru preciznost po broju ljudi i metoda griješi $\pm$ 2 čovjeka. Kod eksperimentalnih mjerenja ne dobivamo dobre rezultate, a razlog je nedovoljan broj detektora.

GNB metoda također je donekle zadovoljavajuća. Nešto jest lošija od DTC metoda, ali ne za puno. Iznenađujuće ima istu ukupnu preciznost kao i DTC za eksperimentalna mjerenja makar je više osjetljiv na preklapanje podataka. GNB metodi bi više odgovaralo statični sustavi to jest sustavi gdje ljudi na primjer uvijek sjede na istim mjestima.

Linearna regresija se pokazala vrlo lošom za oba slučaja pogotovo za eksperimentalni dio gdje metoda daje samo dva rješenja od mogućih osam. Razlog je loša distribucija ulaznih podataka.

Dodaci

Programi

Na linku se nalaze svi programi koji su korišteni za izvedbu ovog diplomskog.

https://drive.google.com/drive/folders/14o6Vy0q1gL5-KTq7AMiLQEBCd27Tlkur?usp=sharing

Opis datoteka:

DTC_COMSOL.ipynb – Decision Tree Classification metoda, numerička simulacija

DTC mjerenja.ipynb – Decision Tree Classification metoda, eksperimentalna mjerenja

GNB_COMSOL.ipynb – Gaussian Naive Bayes metoda, numerička simulacija

GNB mjerenja.ipynb – Gaussian Naive Bayes metoda, eksperimentalna mjerenja

Linear_Models_COMSOL.ipynb – Linearni model, numerička simulacija

Lineat Models mjerenja.ipynb – Linearni model, eksperimentalna mjerenja

Literatura

[1] Decision Trees, https://scikit-learn.org/stable/modules/tree.html, 30.8.2022.

[2] Naive Bayes, https://scikit-learn.org/stable/modules/naive_bayes.html, 30.8.2022.

[3] Linear Models, https://scikit-learn.org/stable/modules/linear_model.html, 31.8.2022

[4] Scikit Learn – Introduction, https://www.tutorialspoint.com/scikit_learn/scikit_learn_quick_guide.htm# , 29.8.2022

[5] Decision Tree, https://www.techopedia.com/definition/28634/decision-tree#:~:text=by%20a%20leaf.-,Classification%20Decision%20Trees,no%3B%20true%2Ffalse), 30.8.2022.

[6] sklearn.model_selection.train_test_split, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html, 5.9.2022.

[7] sklearn.metrics.accuracy_score, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html, 2.9.2022

[8] sklearn.metrics.r2_score, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html, 2.9.2022

[9] White box model, https://www.mdpi.com/1999-4893/13/1/17/htm#:~:text=A%20White-Box%20is%20a,Fuzzy%20Cognitive%20Maps%20%5B12%5D. 30.8.2022

[10] Python, https://en.wikipedia.org/wiki/Python_(programming_language) 26.8.2022

[11] Strojno učenje, https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML 27.8.2022

[12] XOR data set for , https://datascience.stackexchange.com/questions/27960/what-are-examples-for-xor-parity-and-multiplexer-problems-in-decision-tree-lear 30.8.2022