

Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek

Računarstvo usluga i analiza podataka

## SEMINARSKI RAD

„Određivanje rizika komplikacija u trudnoći“

Kristina Dudjak

Osijek, 2022.

## Sadržaj

<b>1. Uvod</b>	<b>1</b>
<b>2. Opis problema</b>	<b>2</b>
2.1. Korišteni podaci	2
2.2. Korišteni postupci strojnog učenja	6
<b>3. Opis programskog rješenja</b>	<b>7</b>
3.1. Model strojnog učenja	7
3.2. Način korištenja API-ja	11
3.3. Klijentska aplikacija	13
<b>4. Zaključak</b>	<b>17</b>
<b>5. Poveznice i literatura</b>	<b>18</b>

## 1. Uvod

Mnoge trudnice koje žive u ruralnim područjima nisu u mogućnosti ići na redovite preglede kod doktora. Samim time njihovo zdravlje ali i zdravlje ploda dovedeno je u rizik. U 2017. godini umrlo je oko 295 000 žena zbog komplikacija tijekom ili nakon trudnoće. Velika većina (94%) smrti dogodila se u siromašnijim zemljama te je većina mogla biti spriječena. [1]

Jedno od mogućih rješenja ovog problema je opisano u ovom projektu. Uporaba strojnog učenja postaje sve popularnija u raznim područjima znanosti, pogotovo u zdravstvu. Korišteni skup podataka dostupan je na UCI repozitoriju, a sastoji se od 1014 unosa prikupljenih iz različitih bolnica i klinika iz ruralnih područja Bangladeša. Podaci su prikupljeni korištenjem IoT tehnologije, tj. pomoću nosivih senzora. Sastoji se od 7 značajki: dob, sistolički tlak, dijastolički tlak, krvni šećer, temperatura, broj otkucaja srca te rizik komplikacija tijekom trudnoće.

Azure Microsoft Machine Learning Studio omogućava treniranje spomenutog skupa podataka. Korišteni tip strojnog učenja je klasifikacija, postupak dohvaćanja uzoraka i smještanje u pripadajuću, unaprijed poznatu kategoriju. Korišteni klasifikator koji daje najbolje rezultate je Multiclass Decision Forest.

Zadovoljavajući model se deploy-ja te koristi kao web servis. Stvorenom API-ju moguće je pristupiti iz razvijene mobilne aplikacije s ciljem predviđanja rizika ovisno o prethodno unesenim značajkama.








## 2. Opis problema

U sljedećim poglavljima opisani su korišteni podaci prikupljeni sa UCI Machine Learning repozitorija te klasifikacijski postupci strojnog učenja.

### 2.1. Korišteni podaci

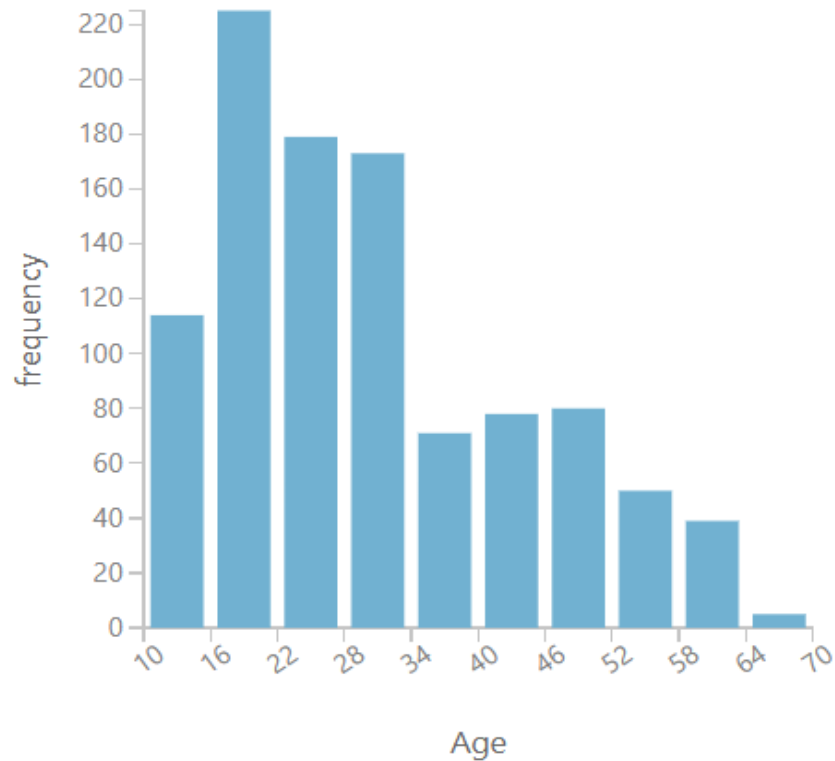
Korišteni skup podataka dostupan je na UCI Machine Learning repozitoriju pod nazivom Maternal Health Risk Data Set. Sadržava 1014 instanci te nema praznih polja, što olakšava izgradnju modela. Predstavljaju ga 7 značajki:

1. Age (dob trudnice)
2. SystolicBP (sistolički tlak) - gornji tlak u mmHg
3. DiastolicBP (dijastolički tlak) - donji tlak u mmHg
4. BS (krvni šećer) - mjereno u mmol/L
5. BodyTemp (temperatura) - mjereno u F
6. HeartRate (broj otkucaja srca u minuti)
7. RiskLevel (rizik komplikacija tijekom trudnoće) - low risk, mid risk ili high risk

Age	SystolicBP	DiastolicBP	BS	BodyTemp	HeartRate	RiskLevel
						
25	130	80	15	98	86	high risk
35	140	90	13	98	70	high risk
29	90	70	8	100	80	high risk
30	140	85	7	98	70	high risk
35	120	60	6.1	98	76	low risk
23	140	80	7.01	98	70	high risk
23	130	70	7.01	98	78	mid risk

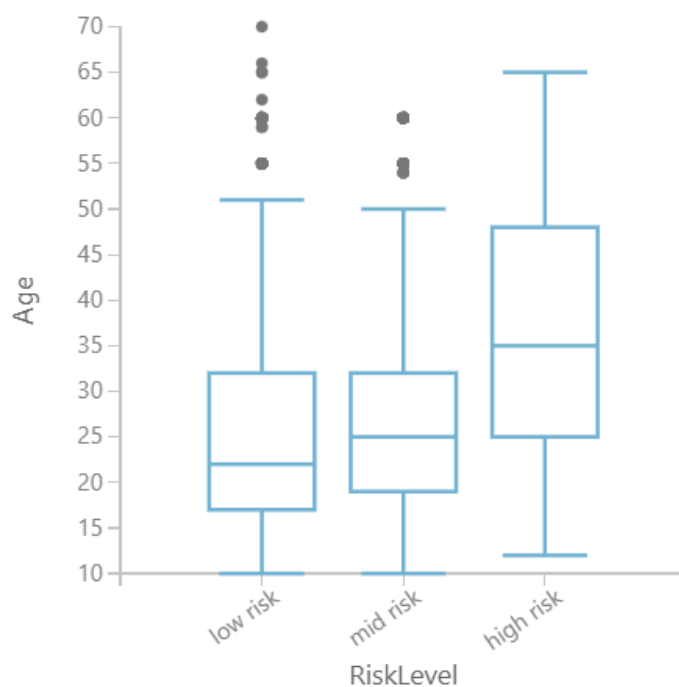
Slika 2.1. Primjer nekoliko unosa podataka i njihov procijenjeni rizik

Na slici 2.1. prikazano je par unosa značajki iz skupa podataka te pripadajuća značajka rizik. Prije izgradnje samog modela, potrebno je upoznati se sa svakom od značajki.



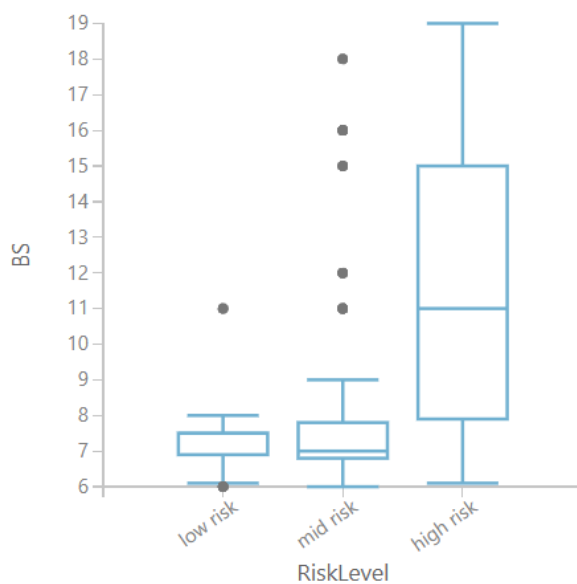
Slika 2.2. Broj unosa ovisno o značajki dob

Na slici 2.2. vidljivo je da je većina ispitanica u životnoj dobi između 16 i 34, zatim između 10 i 16, a ostatak je između 34 i 64 godina. Postoji čak 0.49% njih u životnoj dobi između 64 i 70 godina. Radi se o ruralnim područjima gdje su adolescentne trudnoće češće, ali i trudnoće općenito.



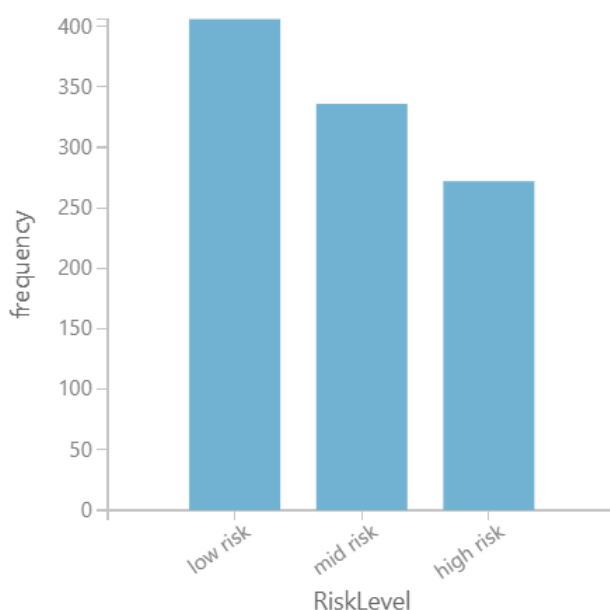
Slika 2.3. Ovisnost životne dobi o riziku

Rizik komplikacija ovisno o dobnim skupinama sa slike 2.3. prikazan je box plotom. Pripadnice niskog rizika su u dobnj skupini između 16 i 32 godine (najčešće 22-23). Srednji rizik prevladava kod dobne skupine između 17 i 32 godine (najčešće 25), a visoki rizik kod dobne skupine između 25 i 50 godina (najčešće 35).



Slika 2.4. Ovisnost krvnog šećera o riziku

Slika 2.4. opisuje nagli porast rizika komplikacija povišenim krvnim šećerom. Isto se može primijetiti i kod visokog sistoličkog i dijastoličkog tlaka.



Slika 2.5. Ovisnost rizika o broju unosa

Sa slike 2.5. vidljivo je da većina ispitanica pripada niskorizičnoj skupini (40%), njih 33% srednjorizičnoj, a 27% visokorizičnoj skupini.

## 2.2. Korišteni postupci strojnog učenja

Iz spomenutog skupa podataka želimo predviđati značajku RiskLevel koja može poprimiti 3 vrijednosti: low risk, mid risk i high risk. Zbog toga je idealni izbor strojnog učenja klasifikacija. Klasifikacija je postupak dohvaćanja uzoraka i njegovo smještanje u pripadajuću kategoriju. Iako zvuči slično grupiranju, razlikuju se po tome što su kod klasifikacije kategorije unaprijed poznate. Pošto nema izostavljenih vrijednosti, ne briše se niti jedna instanca. Također, pošto su svi podaci u sličnom rasponu vrijednosti, nema potrebe za uvođenjem normalizacije podataka.

Postoje mnoge metode obavljanja klasifikacije: decision forest, decision jungle, logistic regression, neural network, i sl. Pošto postoje 3 moguće vrijednosti značajke, potrebno je koristiti navedene metode koje podržavaju više klasa, a ne samo dvije.



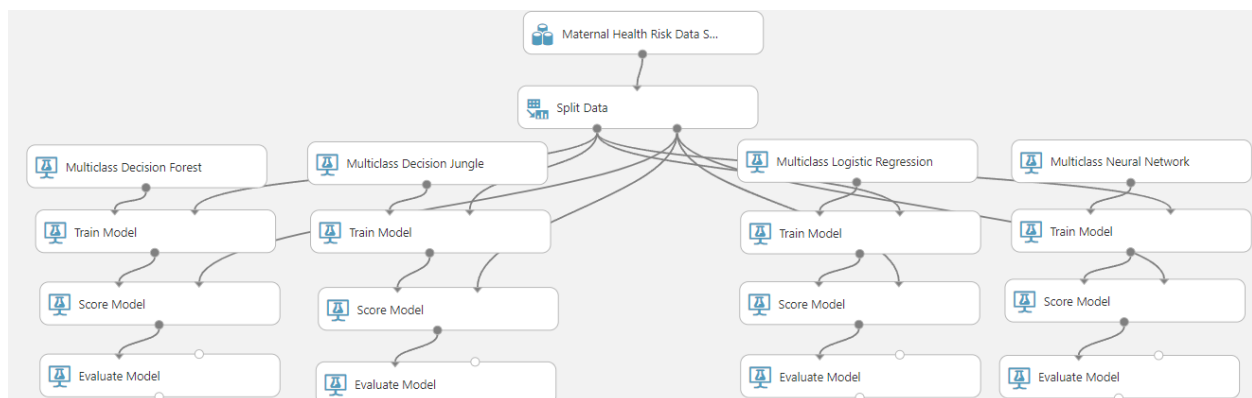
### 3. Opis programskog rješenja

Microsoft Azure Machine Learning Studio omogućava jednostavnu izgradnju modela strojnog učenja. Pomoću njega izvršena je usporedba prethodno spomenutih metoda klasifikacije. Onaj s najboljim rezultatima je deploy-an te korišten u obliku web servisa. Pomoću dostupne API dokumentacije moguće je koristiti POST metodu koja omogućava novi unos podataka i predviđanje rezultata.

Za testiranje i uporabu izgrađenog modela koristi se mobilna aplikacija razvijena u Android Studio-u pomoću programskog jezika Kotlin. Aplikacija omogućava unos potrebnih značajki te na temelju njih izbacuje vrijednost značajke RiskLevel. Za komunikaciju sa API-jem korišten je REST klijent Retrofit koji za upravljanje HTTP zahtjevima koristi OkHttp biblioteku. Za dohvaćanje podataka JSON formata koristi se Gson pretvarač i odgovarajući modeli klasa.

#### 3.1. Model strojnog učenja

Za izgradnju višeklasnog klasifikacijskog modela nude se 4 algoritma: Multiclass Decision Forest, Multiclass Decision Jungle, Multiclass Logistic Regression te Multiclass Neural Network. Prije toga, unutar modula Split Data dodijeljeno je 80% podataka u svrhu treniranja, a 20% u svrhu testiranja. Svakom klasifikacijskom algoritmu dodijeljen je pripadajući Train Model, a unutar svakog je postavljena značajka RiskLevel. Također su postavljeni moduli Score Model i Evaluate Model koji omogućavaju vizualizaciju te usporedbu više modela.



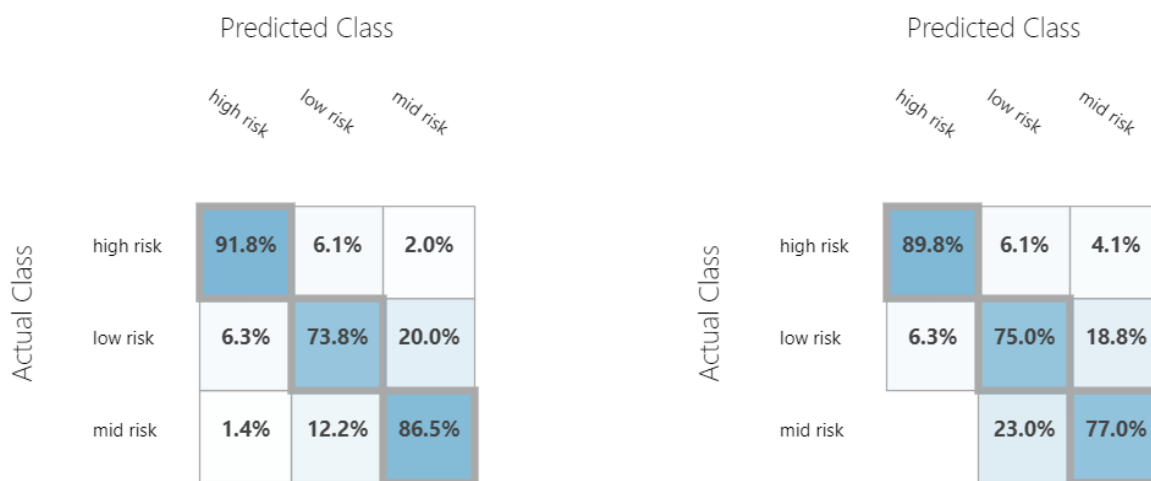
Slika 3.1. Korišteni algoritmi pri izradi modela

Kako bismo odredili koji algoritam je najbolji za izgradnju modela, potrebno ih je usporediti, kao na slici 3.1.

Metrics		Metrics	
Overall accuracy	0.827586	Overall accuracy	0.793103
Average accuracy	0.885057	Average accuracy	0.862069
Micro-averaged precision	0.827586	Micro-averaged precision	0.793103
Macro-averaged precision	0.834487	Macro-averaged precision	0.806076
Micro-averaged recall	0.827586	Micro-averaged recall	0.793103
Macro-averaged recall	0.840244	Macro-averaged recall	0.806076

Slika 3.2. Usporedba Multiclass Decision Forest i Multiclass Decision Jungle

Slika 3.2. prikazuje usporedbu prva dva algoritma, Multiclass Decision Forest te Multiclass Decision Jungle. Prvi algoritam daje sveukupnu točnost od 0.827586%, drugi od 0.793103%.



Slika 3.3. Matrice grešaka Multiclass Decision Forest i Multiclass Decision Jungle

Točnost algoritama moguće je usporediti i matricom grešaka, vidljivo na slici 3.3.

#### Metrics

Overall accuracy	0.596059
Average accuracy	0.730706
Micro-averaged precision	0.596059
Macro-averaged precision	0.633731
Micro-averaged recall	0.596059
Macro-averaged recall	0.58872

#### Metrics

Overall accuracy	0.635468
Average accuracy	0.756979
Micro-averaged precision	0.635468
Macro-averaged precision	0.661465
Micro-averaged recall	0.635468
Macro-averaged recall	0.650375

Slika 3.4. Usporedba Multiclass Logistic Regression i Multiclass Neural Network

Kao što je vidljivo sa rezultata slike 3.4., algoritmi Multiclass Logistic Regression i Multiclass Neural Network daju znatno lošije rezultate od prva dva algoritma.

Konačno, zbog najboljih rezultata izabran je algoritam Multiclass Decision Forest za izradu modela. Nakon namještanja parametara te dodatne usporedbe s ostalim algoritmima, postavljeno je 85% podataka za treniranje, a 15% za testiranje.

#### Metrics

Overall accuracy	0.875
Average accuracy	0.916667
Micro-averaged precision	0.875
Macro-averaged precision	0.875548
Micro-averaged recall	0.875
Macro-averaged recall	0.882143

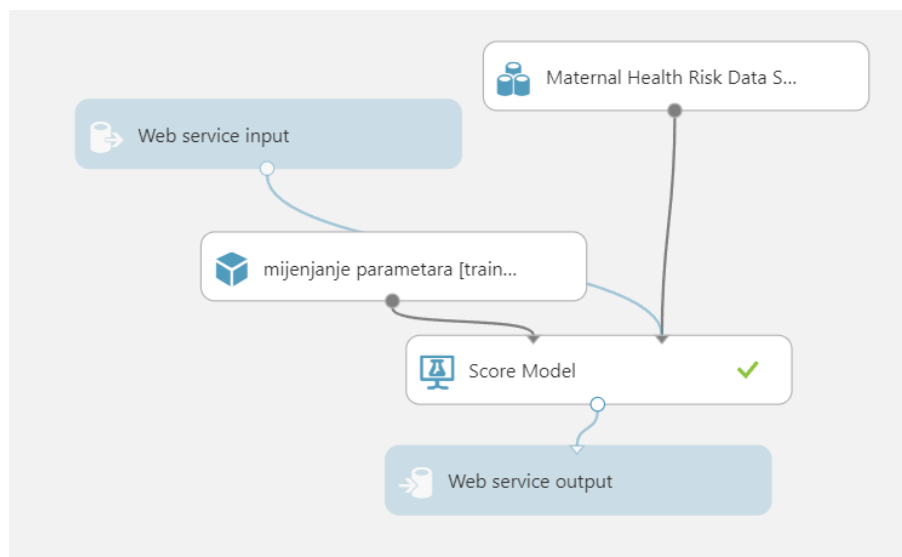
Slika 3.5. Točnost Multiclass Decision Forest

Točnost se popravila sa 0.827586% na 0.875%, vidljivo na slici 3.5. Također, nova matrica grešaka prikazana je na slici 3.6.

		Predicted Class		
		high risk	low risk	mid risk
Actual Class	high risk	95.0%	2.5%	2.5%
	low risk	7.1%	82.1%	10.7%
	mid risk	1.8%	10.7%	87.5%

Slika 3.6. Matrica grešaka Multiclass Decision Forest

Nakon odabira željenog Train modela slijedi izrada web servisa sa slike 3.7.



Slika 3.7. Web servis odabranog modela

## 3.2. Način korištenja API-ja

Uz laku izradu web servisa, Azure nudi i pripadajuću API dokumentaciju za odabrani eksperiment. U njoj se nalaze sve bitne informacije za korištenje i testiranje servisa. Za pristup API-ju koristi se POST zahtjev, a odgovarajući Request URI definiran je u dokumentaciji. Također je potrebno postaviti Request Header koji sadržava API ključ za autorizaciju, Content-Length te Content-Type: application/json.

```
interface APIInterface {  
    @Headers( ...value:  
        "Authorization: Bearer apiKey",  
        "Content-Type: application/json")  
    @POST( value: "execute?api-version=2.0")  
    fun predictRisk(@Body entryData: RequestBody) : Call<ResponseBody>  
}
```

Slika 3.8. Interface *APIInterface*

Na slici 3.8. nalazi se *APIInterface* u kojem su definirana spomenuta zaglavlja, POST zahtjev te funkcija *predictRisk* koja kao parametar sadrži *entryData* tipa *RequestBody* koji ujedno postavljamo kao Body POST zahtjeva. Funkcija vraća *Call<ResponseBody>* što predstavlja odgovor odgovarajućeg HTTP requesta.

```
class ApiService {  
    fun predictRisk(entryData: RequestBody, onResult: (ResponseBody?) -> Unit){  
        val retrofit = ServiceBuilder.buildService(APIInterface::class.java)  
        retrofit.predictRisk(entryData).enqueue(  
            object : Callback<ResponseBody> {  
                override fun onFailure(call: Call<ResponseBody>, t: Throwable) {  
                    onResult(null)  
                }  
                override fun onResponse(call: Call<ResponseBody>, response: Response<ResponseBody>) {  
                    val prediction = response.body()  
                    onResult(prediction)  
                }  
            }  
        )  
    }  
}
```

Slika 3.9. Klasa *ApiService*

Klasa *ApiService* sa slike 3.9. pozivom funkcije *predictRisk* stvara instancu retrofita koja šalje asinkroni request koji vraća *prediction* tipa *ResponseBody*.

```
data class RequestBody(  
    @SerializedName(value: "Inputs") val inputs: Inputs?,  
    @SerializedName(value: "GlobalParameters") val globalParameters: String  
) {  
    companion object {  
        fun create(features: Features): RequestBody {  
            val columnNames = getColumnNames()  
            val values = listOf(getFeatures(features))  
            val input = Input(columnNames, values)  
            val inputs = Inputs(input)  
            return RequestBody(inputs, globalParameters: "")  
        }  
    }  
}
```

Slika 3.10. Klasa *RequestBody*

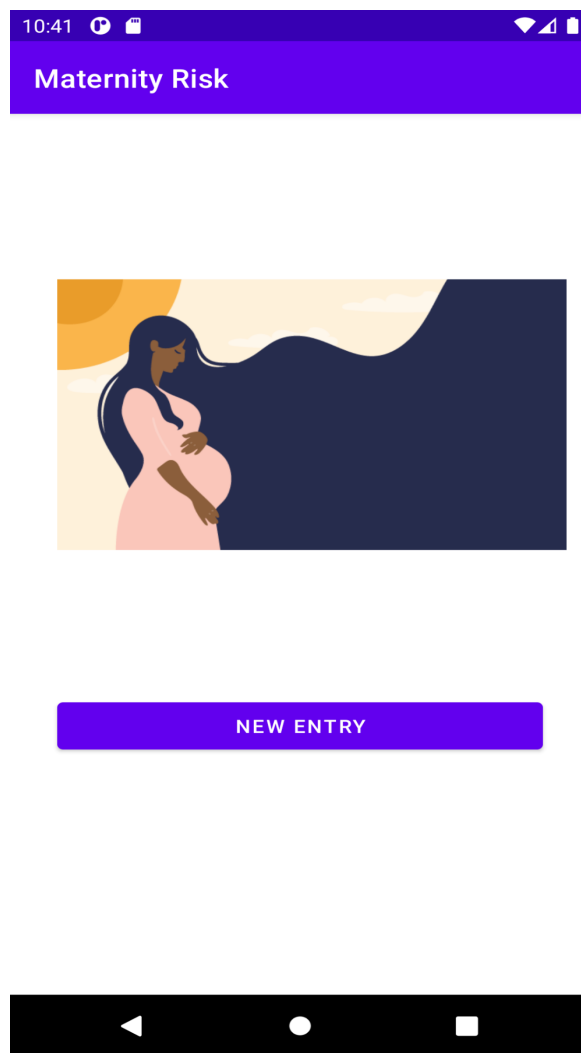
Data klasa *RequestBody* sa slike 3.10. složena je prema Request Body-ju definiranom u API dokumentaciji web servisa. Jedan unos sastoji se od liste *columnNames* koja predstavlja nazive značajki, te *values* koji predstavlja listu liste vrijednosti spomenutih značajki.

```
val apiService = ApiService()  
val age = editAge.toString().toInt()  
val sysBP = editSysBP.toString().toInt()  
val disBP = editDisBP.toString().toInt()  
val bloodSugar = editBS.toString().toDouble()  
val bodyTemp = editBodyTemp.toString().toInt()  
val heartRate = editHeartRate.toString().toInt()  
  
val entryData = RequestBody.create(Features(age, sysBP, disBP, bloodSugar, bodyTemp, heartRate, riskLevel: ""))  
  
apiService.predictRisk(entryData) { it: ResponseBody?   
    if(it?.results?.output?.value?.values != null){  
        val probHigh = it.results.output.value.values[0][7]  
        val probLow = it.results.output.value.values[0][8]  
        val probMid = it.results.output.value.values[0][9]  
        val risk = it.results.output.value.values[0][10]  
  
        val dialog = PredictionDialogFragment()  
        dialog.setTexts(probHigh, probLow, probMid, risk)  
        dialog.show(supportFragmentManager, tag: "customDialog")  
    }  
}
```

Slika 3.11. Isječak koda klase *EntryActivity*

Klasa *EntryActivity* predstavlja Activity u kojem su definirane radnje unosa potrebnih značajki za predviđanje rizika komplikacija tijekom trudnoće. Uneseni podaci predstavljaju značajke potrebne za stvaranje *RequestBody* objekta, vidljivo na slici 3.11. Zatim se poziva funkcija za predviđanje rizika koja vraća odgovarajući *ResponseBody*. Iz njega možemo izvući zanimljive podatke poput predviđene vrijednosti značajke *RiskLevel* te vjerojatnosti pripadnosti low, mid i high skupini rizika. Konačno, poziva se *PredictionDialogFragment()* koji omogućava iskakanje malog prozora sa spomenutim informacijama.

### 3.3. Klijentska aplikacija



Slika 3.12. Početni prozor aplikacije

Na slici 3.12. prikazana je početna stranica aplikacije. Klikom na gumb New Entry, dolazimo do novog prozora u kojem se unose potrebne informacije za predviđanje rizika komplikacija tijekom trudnoće, vidljivo na slici 3.13.

10:41

← New Entry

AGE:  
e.g. 32

SYSTOLIC BP:  
e.g. 120

DIASTOLIC BP:  
e.g. 90

BLOOD SUGAR:  
e.g. 6.9

BODY TEMP:  
e.g. 98

HEART RATE:  
e.g. 70

PREDICT RISK

Slika 3.13. Unos novih podataka

Na slici 3.14. prikazan je primjer unosa informacija. Ako se stisne gumb bez popunjavanja svih polja, aplikacija izbacuje odgovarajuću poruku upozorenja.



The screenshot shows a mobile application interface for a 'New Entry' form. The form is titled 'New Entry' in a blue header bar. Below the header, there are several input fields for medical data: AGE (35), SYSTOLIC BP (100), DIASTOLIC BP (70), BLOOD SUGAR (7), BODY TEMP (98), and HEART RATE (e.g. 70). A red error message 'ALL FIELDS ARE REQUIRED!' is displayed in a white box with a red border, indicating that the HEART RATE field is empty. The bottom of the screen shows a black navigation bar with three icons: a back arrow, a home circle, and a recent apps square.

Field	Value
AGE:	35
SYSTOLIC BP:	100
DIASTOLIC BP:	70
BLOOD SUGAR:	7
BODY TEMP:	98
HEART RATE:	e.g. 70

Slika 3.14. Izostavljanje polja prilikom unosa

Konačno, nakon popunjavanja svih polja, klikom na gumb Predict risk iskače novi prozor koji prikazuje vjerojatnost pripadnosti svakoj od skupina rizika te pripadajuću skupinu rizika, vidljivo na slici 3.15. Klikom na gumb za povratak, mali prozor nestaje.

10:43

← New Entry

AGE:

35

SYSTOLIC BP:

100

DIASTOLIC BP:

70

BLOOD SUGAR:

7

BODY TEMP:

98

HEART RATE:

66

PREDICT RISK

High risk probability: 0.000  
Low risk probability: 0.875  
Mid risk probability: 0.125  
**LOW RISK**

←

Slika 3.15. Prozor sa rezultatima

## 4. Zaključak

Klasifikacijski model izrađen je u Azure ML Studio-u nakon usporedbe odgovarajućih metoda višeklasne klasifikacije: decision forest, decision jungle, logistic regression, neural network. Nakon što je multiclass decision forest dao najbolje rezultate, izrađen je njegov web servis. On osigurava komunikaciju aplikacije sa API-jem što omogućava predavanje potrebnih značajki i predviđanje odabrane značajke RiskLevel: low, mid ili high risk. Mobilna aplikacija koristi Retrofit klijenta za rukovanje POST zahtjevom API servisa.

Uz redovito praćenje promjena tijekom trudnoće, komplikacije i smrtnost tijekom i nakon trudnoće mogu se spriječiti. Ova aplikacija omogućava upravo brz i jednostavan unos zdravstvenih podataka uz koje se može predvidjeti, a naposljetku i spriječiti rizik komplikacija u trudnoći.

## 5. Poveznice i literatura

Programskom je rješenju moguće pristupiti preko:

Github repozitorij: <https://github.com/kristina-dudjak/Maternity-Risk>

ML modeli:

- <https://gallery.cortanaintelligence.com/Experiment/odabrani-Predictive-Exp>
- <https://gallery.cortanaintelligence.com/Experiment/odabrani-Training-Exp>
- <https://gallery.cortanaintelligence.com/Experiment/svi>

Data Set: <https://archive.ics.uci.edu/ml/datasets/Maternal+Health+Risk+Data+Set>

[1] <https://www.who.int/news-room/fact-sheets/detail/maternal-mortality>