

UNIVERZITET „DŽEMAL BIJEDIĆ“ U MOSTARU
FAKULTET INFORMACIJSKIH TEHNOLOGIJA

Akadska 2024./2025. godina

Predmet: Umjetna inteligencija

INTELLIGENTNI AGENT ZA PREDIKCIJU BROJA PACIJENATA U SLUŽBI ZA HITNU MEDICINSKU POMOĆ

Seminarski rad

Profesor:

Prof. dr. Nina Bijedić

Student:

Zaim Mehić, IB210011

Mostar, decembar 2024.

Sadržaj

1. UVOD	3
1.1. Svrha	3
2. IMPLEMENTACIJA.....	4
2.1. Dataset	4
2.2. Python program	5
2.2.1. Korištene biblioteke	5
2.2.2. Učitavanje podataka	5
2.2.3. Treniranje modela	5
2.2.4. API	6
2.3. Web aplikacija	9
3. ZAKLJUČAK.....	10
4. LITERATURA.....	11

1. UVOD

Inteligentni agent za predikciju broja pacijenata u službi za hitnu medicinsku pomoć predviđanje vrši na osnovu historijskih podataka o broju pacijenata po svakom danu pojedinačno vodeći računa o relevantnim podacima koji mogu imati utjecaj na navedeni broj, kao što su temperatura zraka, pritisak, vjetar, dan u sedmici, praznik, itd. Ovaj dokument opisuje implementaciju prve verzije agenta treniranog na podacima Službe za hitnu medicinsku pomoć Doma zdravlja „Stari grad“ u Mostaru i na podacima Federalnog hidrometeorološkog zavoda.

1.1. Svrha

Posljednjih godina svjedočimo rastu potrebe za hitnim medicinskim uslugama u Službama hitne medicinske pomoći, a nažalost kapaciteti velikog broj Službi nisu dovoljni da odgovore na pomenuti rast. Navedeno je rezultovalo prevelikim brojem pacijenata što nadalje utječe ne samo na njihovo zadovoljstvo, nego i na kvalitet liječenja u cjelini.

Izgradnja kapaciteta adekvatnih da odgovore na skoro pa svako opterećenje sistema hitne medicinske pomoći zahtjevalo bi značajna financijska i vremenska ulaganja koja bi za cilj imala porast broja zaposlenih, usavršavanje postojećeg kadra, nadogradnju postojeće infrastrukture i nabavku medicinske opreme. Sve navedeno stvara prostor za dodatnu analizu problema i kreiranje inovativnih rješenja koji za cilj imaju da proizvedu slične efekte uz dosta manja ulaganja.

Svrha PAPER aplikacije je da koristeći jedan od statističkih modela za predikciju kreira pretpostavku o broju pacijenata za određeni dan na osnovu parametara koji imaju vidljiv utjecaj na navedeni broj, te na taj način omogućiti menadžmentu Službe za hitnu medicinsku pomoć da adekvatno rasporedi resurse koje ima na raspolaganju.

Ovakvi alati još uvijek se razvijaju diljem svijeta kako bi rasteretili sistem i osigurali kvalitetnu medicinsku pomoć krucijalnu za dugoročni opstanak navedenog sistema, ali komercijalizacija ovakvih alata još uvijek nije zaživjela ponajviše iz razloga što se radi o veoma osjetljivom području koje daje minimalan prostor za pogreške. Uprkos tome samo je pitanje vremena kada će razvoj umjetne inteligencije i različitih modela za predikciju omogućiti primjenu ovakvih rješenja u realnom svijetu, odnosno donijeti prijeko potrebno olakšanje na veoma opterećeni medicinski sistem.

2. IMPLEMENTACIJA

Završni proizvod ovog seminarskog rada sastoji se od tri dijela:

- a) Dataset-a;
- b) Python programa;
- c) HTML, CSS i JS web aplikacije;

Podaci za vremensku seriju sastoje se od obzervacija prikupljenih vremenom u jednako odvojenim intervalima, a upravo takvi podaci nalaze se u kreiranom Dataset-u. Pošto pored navedenog podaci pokazuju i osobinu sezonalnosti jedini logičan izbor za model koji bi se koristio u ovom slučaju je SARIMAX.

SARIMAX je statistički model dizajniran da uoči i predvidi skrivene uzorke, trendove i sezonalnosti u podacima, a naslanja se na ARIMA model mašinskog učenja.

2.1. Dataset

Podatke o broju pacijenata u Službi za hitnu medicinsku pomoć u periodu od 1.1.2020. godine do 31.5.2024. godine ustupio je Dom zdravlja „Stari grad“ iz Mostara, a podatke o vremenskim prilikama za taj period ustupio je Federalni hidrometeorološki zavod.

Smjer vjetra, godišnja doba, te dani u sedmici prikazanu su na trigonometrijskoj kružnici kako bi se istakla njihova ciklična osobina.

Daljnja poboljšanja ovih podataka mogu se dobiti normalizacijom podataka, te enkodiranjem ostatka kategoričnih varijabli (npr. One-hot enkodiranje)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	date	value	covid	tavg	tmin	tmax	tdiff	minwind	maxwind	windavg	winddiff	pres	sindir	cosdir	seasonsir	seasonco	sinweek	cosweek	holiday
2	2020-01-01		0	8.2	5.2	11.2	6	0.9	8.9	3.704167	8	1017.475	-0.39036	0.920663	0	1	0.974928	-0.22252	1
3	2020-01-02		0	7.4	3.8	11	7.2	1.6	7.4	4.174306	5.8	1018.367	0.471197	0.882028	0	1	0.433884	-0.90097	1
4	2020-01-03		0	8.15	3.8	12.5	8.7	0.8	4.3	2.104861	3.5	1014.288	0.432197	-0.90178	0	1	-0.43388	-0.90097	2
5	2020-01-04		0	6.05	1.8	10.3	8.5	0.5	3.5	1.580556	3	1009.425	-0.827	-0.5622	0	1	-0.97493	-0.22252	0
6	2020-01-05		0	7.15	4	10.3	6.3	0.7	9.2	5.124306	8.5	1012.025	0.418838	0.908061	0	1	-0.78183	0.62349	0
7	2020-01-06		0	4.65	3.1	6.2	3.1	0.6	7.3	3.369444	6.7	1015.725	-0.83634	-0.54821	0	1	0	1	0
8	2020-01-07		0	5.45	0.9	10	9.1	1	6.7	3.519444	5.7	1013.583	0.91321	-0.40749	0	1	0.781831	0.62349	0
9	2020-01-08		0	7.5	4.4	10.6	6.2	1.7	7.1	4.063194	5.4	1016.742	-0.90626	0.422719	0	1	0.974928	-0.22252	0
10	2020-01-09		0	6.05	2.2	9.9	7.7	0.6	4.2	2.065972	3.6	1015.613	-0.76951	-0.63864	0	1	0.433884	-0.90097	0
11	2020-01-10		0	4.3	-1.2	9.8	11	0.4	2.5	0.838889	2.1	1013.117	0.998897	0.046947	0	1	-0.43388	-0.90097	0
12	2020-01-11		0	6.6	-1.2	14.4	15.6	0.4	7.6	3.475	7.2	1013.229	0.997853	-0.06549	0	1	-0.97493	-0.22252	0
13	2020-01-12		0	9.55	7.3	11.8	4.5	1.5	6	3.643056	4.5	1014.429	0.300542	-0.95377	0	1	-0.78183	0.62349	0
14	2020-01-13		0	7.45	3.1	11.8	8.7	0.3	3.7	1.686111	3.4	1014.821	-0.0863	-0.99627	0	1	0	1	0
15	2020-01-14		0	5.55	2.6	8.5	5.9	0	1.9	0.765972	1.9	1014.767	-0.03223	-0.99948	0	1	0.781831	0.62349	0
16	2020-01-15		0	7	1.5	12.5	11	0.3	2.9	1.120139	2.6	1016.833	-0.86376	0.503902	0	1	0.974928	-0.22252	0
17	2020-01-16		0	8.1	0.8	15.4	14.6	0.4	5.1	1.884722	4.7	1015.288	-0.5253	-0.85091	0	1	0.433884	-0.90097	0
18	2020-01-17		0	8.95	4.5	13.4	8.9	0.3	4.7	2.109722	4.4	1012.742	0.912286	0.409553	0	1	-0.43388	-0.90097	0
19	2020-01-18		0	3.95	-0.8	8.7	9.5	0.3	2.7	1.06875	2.4	1010.117	-0.99553	0.094487	0	1	-0.97493	-0.22252	0
20	2020-01-19		0	7.8	5.2	10.4	5.2	0.9	6.2	3.829167	5.3	1012.65	0.870966	0.491343	0	1	-0.78183	0.62349	0
21	2020-01-20		0	6.55	4.5	8.6	4.1	2.3	7.9	5.149306	5.6	1025.538	0.981529	0.191315	0	1	0	1	0
22	2020-01-21		0	6.2	2.8	9.6	6.8	0.7	5.9	3.425694	5.2	1027.133	0.166683	-0.98601	0	1	0.781831	0.62349	0

Slika 1. CSV file - dataset

2.2. Python program

Python program predstavlja osnovu i centralni dio cjelokupne aplikacije, te upotpunosti upravlja procesom vezanim za korišteni model mašinskog učenja, od učitavanja podataka, preko treniranja modela, pa sve do kreiranja predikcija. Ovaj program također služi i kao API na koji se spaja korisnički interfejs kako bi komunicirao sa modelom.

2.2.1. Korištene biblioteke

Flask se koristi za kreiranje API-ja, CORS koristimo kako bismo odobrili komunikaciju između korisničkog sučelja i API-ja, a statsmodel sadrži željeni model mašinskog učenja. Pandas služi za upravljanje datasetom i podacima, a os za upravljanje operativnim sistemom.

```
import os
import pandas as pd
from statsmodels.tsa.statespace.sarimax import SARIMAX
from flask import Flask, request, jsonify
from flask_cors import CORS
```

Slika 2. Korištene biblioteke

2.2.2. Učitavanje podataka

Na samom početku cilj je adekvatno učitati podatke i izvršiti inicijalne prilagodbe podataka, u ovom slučaju prilagodba formata datuma i odabir indeksne kolone.

```
#Učitavanje podataka iz .csv file-a
def data_upload():
    date_format = lambda x: pd.to_datetime(x, format="%Y-%m-%d")
    data = pd.read_csv(csv_file_path, parse_dates=['date'], index_col='date', date_format=date_format)
    data = data.sort_index()
    return data

#Poziv funkcije za učitavanje podataka
data = data_upload()
```

Slika 3. Učitavanje podataka

2.2.3. Treniranje modela

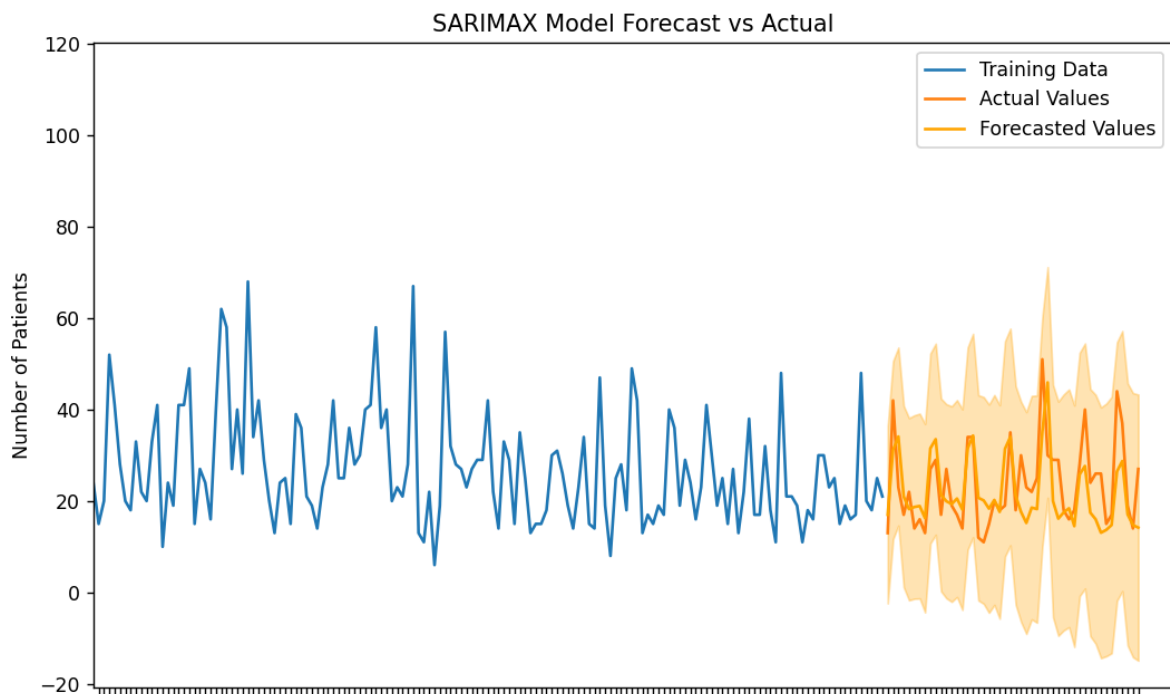
Prilikom inicijalne izrade modela, a u svrhu kontrole kvaliteta podaci su se dijelili na trening i test set podataka u omjeru 80% naprema 20% kako bi se utvrdio kvalitet predikcije na do tada neviđenim podacima. Za produkciju može se koristiti cijeli dataset za treniranje.

Uočena je sedmična sezonalnost što je i uvršteno u parametre u varijabli „seasonal_order“.

```
#Treniranje modela
def train_model():
    global data
    data = data_upload()
    train_size = int(len(data) * 1)
    train = data[:train_size]
    order = (2,1,2)
    seasonal_order = (1,0,2,7)
    train.index = pd.DatetimeIndex(train.index.values, freq="D")
    exog_vars_train = train[['covid', 'tavg', 'tmin', 'tmax', 'tdiff', 'minwind', 'maxwind', 'windavg', 'winddiff', 'pres', 'sindir', 'cosdir', 'seasonsin', 'seasoncos', 'sinweek', 'cosweek', 'holiday']]
    return SARIMAX(train['value'], order=order, seasonal_order=seasonal_order, exog=exog_vars_train, enforce_stationarity=False, enforce_invertibility=False)

#Treniranje modela
sarimax_model = train_model()
results = sarimax_model.fit(maxiter=150)
```

Slika 4. Treniranje modela



Slika 5. Predikcija modela za 48 dana

2.2.4. API

Kreiranje predikcije se radi na način da korisnik proslijedi sve varijable potrebne za kreiranje predviđanja, iste se proslijedjuju modelu koji kreira predikciju. Predikcija i varijable spašavaju se u CSV file gdje čekaju da se unese stvarna vrijednost o broju pacijenata kako bi se izvršilo ponovno treniranje modela.

```

41 #api za kreiranje predviđanja
42 @app.route('/predict', methods=['POST'])
43 def predict():
44     try:
45         json_data = request.get_json()
46         new_data_df = pd.DataFrame(json_data)
47
48         required_columns = ['covid', 'tavg', 'tmin', 'tmax', 'tdiff', 'minwind',
49                             'maxwind', 'windavg', 'winddiff', 'pres', 'sindir',
50                             'cosdir', 'seasonsin', 'seasoncos', 'sinweek', 'cosweek', 'holiday']
51         if not all(col in new_data_df.columns for col in required_columns):
52             raise ValueError(f"Missing required columns. Expected: {required_columns}")
53
54         prediction_input = new_data_df[required_columns]
55         forecast_sarimax = results.get_forecast(steps=len(new_data_df), exog=prediction_input)
56         sarimax_prediction = forecast_sarimax.predicted_mean.tolist()
57
58         document_input = new_data_df[required_columns]
59         document_input['value'] = 0
60         document_input['date'] = new_data_df['date']
61         document_input['pred'] = sarimax_prediction
62         document_input = document_input[['date', 'value'] + required_columns + ['pred']]
63
64         print(document_input)
65         if os.path.exists(csv_file_path):
66             document_input.to_csv(csv_file_path, mode='a', header=False, index=False)
67         return jsonify({"predictions": sarimax_prediction})
68
69     except Exception as e:
70         app.logger.error(f"Error during prediction: {e}")
71         return jsonify({"error": str(e)}), 500
72

```

Slika 6. Kreiranje preikcije

Zabranjeno je raditi nove predikcije prije nego li se popune nedostajuće vrijednosti za već kreirane predikcije. Nakon što se iste unesu vrši se ponovno treniranje i priprema za nova predviđanja.

```

74 @app.route('/real', methods=['POST'])
75 def insert_real_data():
76     try:
77         json_data = request.get_json()
78         new_data_df = pd.DataFrame(json_data)
79
80         required_columns = ['date', 'value']
81         if not all(col in new_data_df.columns for col in required_columns):
82             raise ValueError(f"Missing required columns. Expected: {required_columns}")
83
84         df = pd.read_csv(csv_file_path)
85
86         for x in json_data:
87             df.loc[df['date']==x['date'], 'value'] = x['value']
88
89         df.to_csv(csv_file_path, index=False)
90         global sarimax_model
91         sarimax_model = train_model()
92         global results
93         results = sarimax_model.fit(maxiter=150)
94         return jsonify({"response": str("Ok")}), 200
95     except Exception as e:
96         app.logger.error(f"Error during prediction: {e}")
97         return jsonify({"error": str(e)}), 500
98
99

```

Slika 7. Unos stvarnih podataka

Web aplikacija predstavlja korisnički interfejs između korisnika i agenta, a u okviru seminarskog rada služi za jednostavan prikaz mogućnosti inteligentnog agenta.

Da bismo prilikom svakog pokretanja korisničkog interfejsa otvorili odgovarajući ekran, potrebno je provjeriti postoji li instanci koje nemaju unesene stvarne vrijednosti o broju pacijenata.

```

100 @app.route('/check_missing_values', methods=['GET'])
101 def check_missing_values():
102     try:
103         data = pd.read_csv(csv_file_path, parse_dates=['date'])
104         missing_value_records = data[data['value']==0].sort_values(by='date')
105
106         if not missing_value_records.empty:
107             missing_records_list = missing_value_records.to_dict(orient='records')
108             for x in missing_records_list:
109                 x['date'] = x['date'].strftime('%Y-%m-%d')
110             return jsonify({"missing_records": missing_records_list})
111
112         last_dates = []
113         for i in range(7):
114             last_dates.append((pd.to_datetime(data['date'].max(), format='%Y-%m-%d')+ pd.Timedelta(days=i+1)).strftime('%Y-%m-%d'))
115         return jsonify({"missing_records": {}, "message": "No missing values found", "last_dates": last_dates})
116
117     except Exception as e:
118         app.logger.error(f"Error in check_missing_values endpoint: {e}")
119         return jsonify({"error": str(e)}), 500
120

```

Slika 8. Provjera vrijednosti

2.3. Web aplikacija

Korisnički interfejs omogućava korisniku da unese parametre za najviše 7 narednih dana i dobije predikciju o broju pacijenata za navedeni period.

The screenshot displays the 'Paper - Weekly Prediction' web application interface. It features seven columns, each representing a day from Day 1 to Day 7. Each column contains a date field (e.g., '22-Jun-2024' for Day 1), a 'Covid' dropdown menu, and input fields for 'Minimum Temperature', 'Maximum Temperature', 'Minimum Wind Speed', 'Maximum Wind Speed', 'Pressure', and 'Wind Direction'. Additionally, there are 'Season' and 'Day of the Week' dropdown menus, and a 'Holiday' dropdown menu. At the bottom of each column, there is a 'Prediction: N/A' label and an 'Actual' input field with a placeholder 'Enter actual value'.

Slika 9. Unos varijabli za predikciju

Nakon što se kreira predikcija, od korisnika se traži da unese stvarni broj pacijenata na navedene dane.

The screenshot displays the 'Paper - Weekly Prediction' web application interface after the prediction phase. It features seven columns, each representing a day from Day 1 to Day 7. Each column contains a date field (e.g., '23-Jun-2024' for Day 1), a 'Prediction' value (e.g., '32' for Day 1), and an 'Actual' input field with a placeholder 'Enter actual value'. A blue button labeled 'Save Actual Data and Retrain' is located at the bottom center of the interface.

Slika 10. Unos stvarnog broja pacijenata

Na ovaj način je osigurano da se unos podataka vrši kontinuirano što će u konačnici spriječiti greške prilikom treniranja modela.

3. ZAKLJUČAK

Vrijednost srednje kvadratne greške je prilikom posljednjeg testiranja i prilikom predikcije za narednih 48 dana iznosila 59.7, a vrijednost srednje apsolutne greške iznosila je 6.1.

U skladu sa navedenim vrijednostima možemo zaključiti da je model uspio otkriti uzorke i sezonalnosti, te da može biti odličan alat koji Službi za hitnu medicinsku pomoć može dati okvirni prikaz onoga što ih može očekivati u narednom periodu.

U gornjim paragrafima su izdvojena određena poboljšanja koja se mogu primijeniti na trenutni dataset, te je moguće da će se u nekoj od narednih iteracija ovog projekta srednja kvadratna i apsolutna greška smanjiti u određenoj mjeri.

4. LITERATURA

1. Hye Jin Kam, Jin Ok Sung, Rae Woong Park (2010), Prediction of Daily Patient Numbers for a Regional Emergency Medical Center using Time Series Analysis, National Library of Medicine
2. Farzad Karami, Yujie Zhang, Mario A. Rotea, Federico Bernardoni, Stefano Leonardi (2021), Real-time Wind Direction Estimation using Machine Learning on Operational Wind Farm Data, IEEE Conference on Decision and Control