

Prognoza vremena na bazi UI za lokalno područje

Baranić, Florijan

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Polytechnic of Međimurje in Čakovec / Međimursko veleučilište u Čakovcu**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:110:592680>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-03**



Repository / Repozitorij:

[Polytechnic of Međimurje in Čakovec Repository -
Polytechnic of Međimurje Undergraduate and
Graduate Theses Repository](#)



MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

FLORIЈAN BARANIĆ

**PROGNOZA VREMENA NA BAZI UI ZA LOKALNO
PODRUČJE**

ZAVRŠNI RAD

Mentor:

mr. sc. Željko Knok, v. pred.

Čakovec, srpanj 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
STRUČNI PRIJEDIPLOMSKI STUDIJ RAČUNARSTVO

FLORIЈAN BARANIĆ

**WEATHER FORECAST ON A UI BASE FOR THE
LOCAL AREA**

ZAVRŠNI RAD

Mentor:

mr. sc. Željko Knok, v. pred.

Čakovec, srpanj 2023.

MEĐIMURSKO VELEUČILIŠTE U ČAKOVCU
ODBOR ZA ZAVRŠNI RAD

Čakovec, 2. veljače 2023.

država: **Republika Hrvatska**
Predmet: **Baze podataka II-izborni**

ZAVRŠNI ZADATAK br. 2022-RAČ-R-48

Pristupnik: **Florijan Baranić (0313025632)**
Studij: **Redoviti preddiplomski stručni studij Računarstvo**
Smjer: **Programsko inženjerstvo**

Zadatak: **Prognoza vremena na bazi UI za lokalno područje**

Opis zadatka:

Koristeći web scraping odabrati nekoliko meto-stanica (min. 3) na lokalnom području, pratiti njihove podatke kroz priod od mjesec dana. Na osnovu prikupljenih podataka pomoću UI kreirati neuronsku mrežu koja će za navedeno razdoblje praćenja omogućiti prognozu za isto vremensko razdoblje iduće kalendarske godine. Uz pomoć Anaconda i dodatnom instalacijom potrebnih alata kao što su programski jezik Python i moduli-biblioteke koje su sastavni dio programskog jezika, a koriste se za izradu rada.

Alati:

Visual Studio Code, Web scraping, Anaconda, Python, potrebni moduli-biblioteke

Zadatak uručen pristupniku: 2. veljače 2023.

Rok za predaju rada: 20. rujna 2023.

Mentor:



mr. sc. Željko Knok, v. pred.

Predsjednik povjerenstva za
završni ispit:

ZAHVALA

Želim se posebno zahvaliti mentoru na pomoći i usmjeravanju tijekom izrade ovog završnog rada. Zahvaljujem se i svim profesorima na Međimurskom veleučilištu u Čakovcu koji su mi jako puno pomogli i zaslužni su u stjecanju mojeg znanja kroz studij.

SAŽETAK

Svaka tehnologija koja se koristi za izradu aplikacije je ukratko opisana. Predviđanje vremena je vrlo složen proces, a povratne informacije o vremenu i na lokalnom području mogu biti od koristi za mnoge ljude.

Koristeći *web scraping*, autor je odabrao 3 *meteo-stanice* na lokalnom području, a to su: Čakovec, Varaždin i Koprivnica. Prate se njihovi *podaci* u periodu od 30 dana i spremaju se u excel datoteku. Na osnovu prikupljenih podataka pomoću *UI*, kreira se *neuronska mreža* koja će za navedeno razdoblje praćenja omogućiti prognozu za isto vremensko razdoblje iduće kalendarske godine.

Autor je odabrao i koristio specifično *Mlp* (Multilayer Percepton) neuronsku mrežu za prognoziranje vremenskih uvjeta u budućnosti. Trenirajući i testirajući, autor je dobio najbolje rezultate s ovom neuronskom mrežom i iz tog razloga ju je odabrao, radio je u *Jupyter Notebook-u* koji je dio *Anaconda* okruženja.

Anaconda okruženje omogućuje jednostavno upravljanje i instalaciju potrebnih alata i biblioteka, kao što je programski jezik *Python*. Kroz rad su korišteni različiti moduli i biblioteke koje su sastavni dio programskog jezika *Python* i autor je opisao svaku od njih.

Uzimajući u obzir sve navedeno, pokazuje se važnost umjetne inteligencije, strojnog učenja, neuronskih mreža i *web scraping* tehnika u području prognoziranja vremena. Kombinacija ovih tehnologija omogućuje preciznije i relevantnije prognoze vremena na lokalnom području.

Budućnost tih područja donosi veliki potencijal za daljnji razvoj i primjenu naprednih tehnika kako bi se još poboljšala preciznost *prognoza* i omogućilo bolje razumijevanje vremenskih uvjeta.

Ovaj završni rad pruža temelje za daljnja istraživanja i razvoj vremenskih *prognoza* temeljenih na umjetnoj inteligenciji i strojnom učenju.

Ključne riječi: *web scraping*, *meteo-stanice*, *UI*, *neuronska mreža*, *prognoza*, *Anaconda*, *Python*.

Sadržaj

1. UVOD.....	1
2. STROJNO UČENJE.....	2
2.1. Funkcioniranje strojnog učenja	2
2.2. Modeli strojnog učenja.....	3
2.3. Algoritmi strojnog učenja.....	4
2.3.1. Neuronske mreže	4
2.3.2. Linearna regresija.....	5
2.3.3. Logistička regresija.....	5
2.3.4. Grupiranje	5
2.3.5. Stabla odlučivanja.....	5
2.3.6. Nasumične šume	5
3. KORIŠTENI ALATI I TEHNOLOGIJE	6
3.1. Web scraping	6
3.1.1. Proces Web scrapinga	7
3.2. Neuronska mreža	7
3.2.1. Vrste neuronskih mreža.....	8
3.3. Anaconda.....	10
3.3.1. Jupyter notebook.....	11
3.4. Python	13
4. KORIŠTENE PYTHON BIBLIOTEKE	14
4.1 Pandas	14
4.1.1. NumPy.....	14
4.1.2. Matplotlib	15
4.1.3. Sklearn.....	15

4.1.4. Tkinter	15
4.1.5. DateTime	15
4.1.6. Pillow	15
4.1.7. PyQt5.....	16
4.1.8. Requests.....	16
4.1.9. BeautifulSoup	16
4.2.0. Csv	16
5.SKUPLJANJE PODATAKA PUTEM WEB SCRAPING-A ZA APLIKACIJU	17
5.1. Automatsko dohvaćanje podataka	17
5.1.1. Spremanje dohvaćenih podataka.	19
6.FUNKCIONALNOST APLIKACIJE	20
6.1. Glavno sučelje	21
6.2. Treniranje i testiranje sučelje.....	23
6.2.1. Funkcije treniranje i testiranje sučelja.....	25
6.2. Statistika sučelje	29
6.2. Funkcije statistika sučelja.....	31
6.3. Vrijeme sučelje	34
6.3.1. Funkcije vrijeme sučelja.....	35
6.3. Vremenska prognoza sučelje	38
6.3.1. Vremenska prognoza sučelje funkcije	39
6.4. Treniranje-testiranje-pojedinačno	46
6.4.1. Treniranje- testiranje-pojedinačno sučelje funkcije.....	47
6.5. Funkcije sučelja	47
7.ZAKLJUČAK	49
8. LITERATURA	50
9.POPIS SLIKA	52

1. UVOD

Vremenska prognoza je rezultat procesa predviđanja (prognoze) stanja atmosfere za određeno mjesto i vrijeme pomoću znanosti (meteorologije) i tehnologije. Vremenska prognoza uvijek se odnosi na vrijeme u budućnosti, vremenske prognoze za prošlost nema. [12]

Vremenska prognoza je važna za mnoge ljude i industrije i ima široku primjenu u različitim područjima, kao što su poljoprivreda, transport, turizam i mnoge druge. Može uključivati informacije o oborinama, tlaku zraka, smjeru i brzini vjetrova, temperaturi, vlažnosti zraka i svim drugim pojavama.

U posljednjih nekoliko desetljeća umjetna inteligencija i strojno učenje napravili su revoluciju u prognoziranju vremena. Kako strojno učenje napreduje i sve više vremenskih modela ga počinje integrirati, vremenska prognoza postat će sve točnija. [13]

Cilj završnog rada je kreirati neuronsku mrežu, trenirati ju i testirati, izraditi predikciju vremena i prikazati ju na korisničkom sučelju pomoću grafova i tablica, koja će olakšati svim industrijama i ljudima. Točne prognoze mogu pomoći ljudima u planiranju svojih aktivnosti, kao što su putovanja, aktivnosti na otvorenom, sigurnost da se pripreme za vremenske nepogode, kao što su oluje ili snježne mećave, i zdravlja zbog zagađenog zraka i povišene temperature, a industrijama da planiraju svoje aktivnosti, kao što su sadnja, berba, ribolov, izgradnja i održavanje infrastrukture i drugo. To može pomoći u poboljšanju produktivnosti, smanjenju troškova i povećanju profita.

Autor je kroz 30 dana skupljao podatke o: danu, mjestu, trenutnoj temperaturi, oborinama, vlazi, vjetru, trenutnom vremenu, najvećoj temperaturi, najmanjoj temperaturi. Web stranica iz koje je prikupljao podatke pomoću web scraping-a: google vrijeme (weather.com).

Izrađena predikcija ima mogućnost pregleda 3 lokalna područja. Korisnik može odabrati 3 excel datoteke s podacima za navedena lokalna područja.

2. STROJNO UČENJE

Strojno učenje grana je umjetne inteligencije (AI) i računalne znanosti koja se fokusira na upotrebu podataka i algoritama za oponašanje načina na koji ljudi uče, postupno poboljšavajući njegovu točnost.[9]

Metode strojnog učenja omogućuju računalima da rade autonomno bez eksplicitnog programiranja. ML aplikacije se hrane novim podacima i mogu samostalno učiti, rasti, razvijati se i prilagođavati. [10]

Strojno učenje izvlači pronicljive informacije iz velikih količina podataka korištenjem algoritama za prepoznavanje uzoraka i učenje u iterativnom procesu. ML algoritmi koriste računalne metode za učenje izravno iz podataka, umjesto da se oslanjaju na bilo koju unaprijed određenu jednadžbu koja može poslužiti kao model. [10]

Iako strojno učenje nije novi koncept – potječe iz Drugog svjetskog rata kada je korišten Enigma Machine – mogućnost automatske primjene složenih matematičkih izračuna na sve veće količine i raznolikost dostupnih podataka relativno je novi razvoj. Danas, s porastom velikih podataka, strojno učenje postalo je ključno za rješavanje problema u brojnim područjima.[10]

Algoritmi strojnog učenja obično se izrađuju pomoću okvira koji ubrzavaju razvoj rješenja, kao što su TensorFlow i PyTorch.[9]

2.1. Funkcioniranje strojnog učenja

1. Proces odlučivanja: Općenito, algoritmi strojnog učenja koriste se za predviđanje ili klasifikaciju. Na temelju nekih ulaznih podataka, koji mogu biti označeni ili neoznačeni, algoritam će proizvesti procjenu uzorka u podacima. [9]
2. Funkcija pogreške: Funkcija pogreške procjenjuje predviđanje modela. Ako postoje poznati primjeri, funkcija pogreške može napraviti usporedbu za procjenu točnosti. [9]
3. Proces optimizacije modela: Ako se model može bolje uklopiti u podatkovne točke u skupu za obuku, tada se težine prilagođavaju kako bi se smanjila razlika između poznatog primjera i procjene modela. Algoritam će ponoviti ovaj proces „procjene i optimizacije”, autonomno ažurirajući težine sve dok se ne postigne prag točnosti. [9]

2.2. Modeli strojnog učenja

Algoritmi strojnog učenja mogu se trenirati na mnogo načina, a svaka metoda ima svoje prednosti i mane. Na temelju ovih metoda i načina učenja, strojno učenje se općenito kategorizira u četiri glavne vrste:

2.2.1. Nadzirano strojno učenje

Ova vrsta ML-a uključuje nadzor, gdje se strojevi obučavaju na označenim skupovima podataka i omogućava im se predviđanje rezultata na temelju pružene obuke. Označeni skup podataka navodi da su neki ulazni i izlazni parametri već mapirani. Dakle, stroj se trenira s ulazom i odgovarajućim izlazom. Napravljen je uređaj za predviđanje ishoda pomoću skupa testnih podataka u sljedećim fazama. [10]

Na primjer, razmotrite ulazni skup podataka slika papige i vrane. U početku se stroj obučava da razumije slike, uključujući boju, oči, oblik i veličinu papige i vrane. Nakon treninga daje se ulazna slika papige, a od stroja se očekuje da identificira objekt i predvidi rezultat. Istrenirani stroj provjerava različite značajke objekta, kao što su boja, oči, oblik itd., na ulaznoj slici, kako bi napravio konačno predviđanje. Ovo je proces identifikacije objekta u nadziranom strojnom učenju. [10]

Primarni cilj tehnike nadziranog učenja je mapiranje ulazne varijable (a) s izlaznom varijablom (b). [10]

2.2.2. Strojno učenje bez nadzora

Učenje bez nadzora odnosi se na tehniku učenja koja je lišena nadzora. Ovdje se stroj obučava korištenjem neoznačenog skupa podataka i omogućeno mu je predviđanje rezultata bez ikakvog nadzora. Algoritam za učenje bez nadzora ima za cilj grupirati nerazvrstani skup podataka na temelju sličnosti, razlika i uzoraka ulaza. [10]

Na primjer, razmotrite ulazni skup podataka slika posuda pune voća. Ovdje slike nisu poznate modelu strojnog učenja. Kada unosimo skup podataka u ML model, zadatak modela je identificirati uzorak objekata, kao što su boja, oblik ili razlike koje se vide na ulaznim slikama i kategorizirati ih. Nakon kategorizacije, stroj predviđa izlaz dok se testira s testnim skupom podataka. [10]

2.2.3. Polunadzirano učenje

Polunadzirano učenje uključuje karakteristike i nadziranog i nenadziranog strojnog učenja. Koristi kombinaciju označenih i neoznačenih skupova podataka za obuku svojih algoritama. Koristeći obje vrste skupova podataka, polunadzirano učenje prevladava nedostatke gore navedenih opcija. [10]

Razmotrimo primjer studenta. Učenje, kada student uči koncept pod nadzorom nastavnika na fakultetu, naziva se učenje pod nadzorom. U učenju bez nadzora, učenik samostalno uči isti koncept kod kuće bez vodstva učitelja. U međuvremenu, učenje, u kojem student revidira koncept nakon učenja pod vodstvom nastavnika na fakultetu, je polunadzirani oblik učenja. [10]

2.2.4. Učenje s potkrepljenjem

Učenje s potkrepljenjem proces je temeljen na povratnim informacijama. Ovdje AI komponenta automatski provjerava svoje okruženje metodom hit & trial, poduzima radnje, uči iz iskustava i poboljšava performanse. Komponenta se nagrađuje za svaku dobru akciju i kažnjava za svaki pogrešan potez. Dakle, komponenta učenja s potkrepljenjem ima za cilj maksimizirati nagrade izvođenjem dobrih radnji. [10]

Za razliku od nadziranog učenja, učenju s potkrepljenjem nedostaju označeni podaci, a agenti uče samo kroz iskustva. Razmotrite video igre. Ovdje igra specificira okruženje, a svaki potez agenta pojačanja definira njegovo stanje. Agent ima pravo primiti povratne informacije putem kazni i nagrada, čime utječe na ukupni rezultat igre. Krajnji cilj agenta je postizanje visokog rezultata. [10]

Učenje s pojačanjem primjenjuje se u različitim područjima, kao što su teorija igara, teorija informacija i sustavi s više agenata.[10]

2.3. Algoritmi strojnog učenja

Često se koristi niz algoritama strojnog učenja. To uključuje:

2.3.1. Neuronske mreže

Neuronske mreže simuliraju način na koji ljudski mozak radi, s ogromnim brojem povezanih čvorova za obradu. Neuronske mreže su dobre u prepoznavanju uzoraka i igraju važnu ulogu u

aplikacijama uključujući prevođenje prirodnog jezika, prepoznavanje slika, prepoznavanje govora i stvaranje slika. [10]

2.3.2. Linearna regresija

Ovaj se algoritam koristi za predviđanje numeričkih vrijednosti, na temelju linearnog odnosa između različitih vrijednosti. Na primjer, tehnika se može koristiti za predviđanje cijena kuća na temelju povijesnih podataka za to područje. [10]

2.3.3. Logistička regresija

Ovaj algoritam nadziranog učenja daje predviđanja za varijable kategoričkih odgovora, kao što su odgovori 'da/ne' na pitanja. Može se koristiti za aplikacije, kao što je klasificiranje neželjene pošte i kontrola kvalitete na proizvodnoj liniji. [10]

2.3.4. Grupiranje

Koristeći učenje bez nadzora, algoritmi grupiranja mogu identificirati uzorke u podacima kako bi se oni mogli grupirati. Računala mogu pomoći znanstvenicima koji se bave podacima identificirajući razlike između stavki podataka koje su ljudi previdjeli. [10]

2.3.5. Stabla odlučivanja

Stabla odlučivanja mogu se koristiti i za predviđanje numeričkih vrijednosti (regresija) i za klasifikaciju podataka u kategorije. Stabla odlučivanja koriste niz grananja povezanih odluka koje se mogu prikazati dijagramom stabla. Jedna od prednosti stabala odlučivanja je ta što ih je lako potvrditi i revidirati, za razliku od crne kutije neuronske mreže. [10]

2.3.6. Nasumične šume

U nasumičnoj šumi, algoritam strojnog učenja predviđa vrijednost ili kategoriju kombiniranjem rezultata određenog broja stabala odlučivanja. [10]

3. KORIŠTENI ALATI I TEHNOLOGIJE

U ovom poglavlju objašnjene su tehnologije koje se koriste za izradu aplikacije. Poglavlje objašnjava korištenje tehnike kako bi se osigurala sigurnost aplikacije.

3.1. Web scraping

Web scraping je proces prikupljanja strukturiranih web podataka na automatiziran način. Poznato je i kao ekstrakcija web podataka, koriste ga ljudi i tvrtke koji žele koristiti javno dostupne web podatke. Postoje dva načina izdvajanja podataka s web stranica, tehnika ručnog izdvajanja i automatizirana tehnika izdvajanja. Kod ručnog izdvajanja koristi se ručno kopiranje i lijepljenje sadržaja, u ovoj tehnici oduzima se puno vremena i ponavlja se, dok se kod automatizirane tehnike koristi automatsko izdvajanje podataka sa stranica na temelju zahtjeva korisnika.

Neki od glavnih slučajeva korištenja web scrapinga uključuju istraživanje tržišta, novinarstvo, akademska istraživanja, izrada aplikacije. Postoji mnogo razloga za korištenje web scrapinga, a njegova popularnost raste iz godine u godinu kako se sve više podataka pojavljuje na internetu. Na nekim web-lokacijama podaci su dostupni za jednostavno preuzimanje u CSV ili JSON formatu, ali u nekim slučajevima to nije moguće i za to je potreban web scraping .[1]

Alat za struganje ili strugač web-mjesta koristi se kao dio procesa struganja weba za postavljanje HTTP zahtjeva na ciljanom web-mjestu i izdvajanje web-podataka sa stranice. Raščlanjuje sadržaj koji je javno dostupan i vidljiv korisnicima, a poslužitelj ga prikazuje kao HTML. Ponekad također upućuje zahtjeve internim programskim sučeljima aplikacija (API) za povezane podatke – poput cijena proizvoda ili pojedinosti o kontaktu – koji su pohranjeni u bazi podataka i isporučeni pregledniku putem HTTP zahtjeva, on je softverski program dizajniran za izdvajanje (ili 'struganje weba') relevantnih podataka s web stranica.

Postoje razne vrste web strugača i alata za ekstrakciju podataka, korisni su svima koji pokušavaju prikupiti određene podatke s web stranica jer korisniku pružaju strukturirane podatke izvlačeći podatke s brojnih web stranica.[1]

3.1.1. Proces Web scrapinga

1. Zahtjev- Identificira se ciljna web stranica i podaci koji se žele prikupiti. Program koristi naredbu „GET“ za povlačenje podataka sa stranice koju smo odabrali.
2. Odabir alata za web scraping - Postoje mnogi alati, kao što je BeautifulSoup, Python biblioteka za izvlačenje podataka iz HTML i XML datoteka.
3. Prikaz - Pronalaženje ciljanih podataka na web stranici, to se može učiniti ručno ili pomoću selektora.
4. Izvoz podataka - Podaci se mogu spremiti u različite formate, kao što su CSV, JSON, Excel.

```
1 import requests
2 from bs4 import BeautifulSoup
3 import csv
4 url = "https://www.google.com/search?q=vrijeme+Koprivnica"
5 headers = {
6     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/
7 }
8 page = requests.get(url, headers=headers)
9
10 soup = BeautifulSoup(page.content, "html.parser")
11
12 dan=soup.find("div",attrs={"id":"wob_dts"}).text
13 mjesto=soup.find("span",attrs={"class":"BBwThe"}).text
14 temp = soup.find("span", attrs={"id": "wob_tm"}).text
15 oborine=soup.find("span",attrs={"id":"wob_pp"}).text
16 vlaga=soup.find("span",attrs={"id":"wob_hm"}).text
17 vjetar=soup.find("span",attrs={"id":"wob_ws"}).text
18 trenut=soup.find("span",attrs={"id":"wob_dc"}).text
19 max=soup.find("div",attrs={"class":"gNCp2e"}).find_next('span', attrs={'class': 'wob_t'}).text
20 min=soup.find("div",attrs={"class":"QrNVmd ZXcv8e"}).find_next('span', attrs={'class': 'wob_t'}).text
21 print(f"Dan: {dan}")
22 dan=soup.find("div",attrs={"id":"wob_dts"}).text
23 print(f"Mjesto: {mjesto}")
24 print(f"Temperatura: {temp}")
25 print(f"Oborine: {oborine}")
26 print(f"Vlaga: {vlaga}")
27 print(f"Vjetar: {vjetar}")
28 print(f"Trenutno vrijeme: {trenut}")
29 print(f"Najveća temperatura: {max}")
30 print(f"Najmanja temperatura: {min}")
31
32
```

Slika 1. Prikaz procesa web scrapinga.

Izvor: Autor

3.2. Neuronska mreža

Neuronska mreža (također nazvana umjetna neuronska mreža) prilagodljivi je sustav koji uči pomoću međusobno povezanih čvorova ili neurona u slojevitoj strukturi koja nalikuje ljudskom mozgu. Neuronska mreža može učiti iz podataka, tako da se može osposobiti za prepoznavanje obrazaca, klasificiranje podataka i predviđanje budućih događaja.

Neuronska mreža rastavlja ulaz u slojeve apstrakcije. Može se uvježbati korištenjem mnogih primjera kako bi prepoznala obrasce u govoru ili slikama, kao što to radi ljudski mozak. Njegovo ponašanje definirano je načinom na koji su njegovi pojedinačni elementi povezani i snagom ili težinom tih veza. Te se težine automatski prilagođavaju tijekom treninga u skladu s određenim pravilom učenja sve dok umjetna neuronska mreža ispravno ne izvede željeni zadatak. [2]

Neuronske mreže vrsta su pristupa strojnom učenju inspiriranog načinom na koji neuroni signaliziraju jedni drugima u ljudskom mozgu. Ovi su pristupi ključna tehnologija koja pokreće inovacije u naprednim sustavima. [2]

Inspirirana biološkim živčanim sustavima, neuronska mreža kombinira nekoliko slojeva obrade, koristeći jednostavne elemente koji rade paralelno. Mreža se sastoji od ulaznog sloja, jednog ili više skrivenih slojeva i izlaznog sloja. U svakom sloju postoji nekoliko čvorova ili neurona. [2]

3.2.1. Vrste neuronskih mreža

Feedforward neuronska mreža jedna je od jednostavnijih vrsta neuronskih mreža. Prenosi informacije u jednom smjeru kroz ulazne čvorove. Te se informacije nastavljaju obrađivati u jednom smjeru sve dok ne dosegnu izlazni način. Feed-forward neuronske mreže mogu imati skrivene slojeve za funkcionalnost, a ova vrsta se najčešće koristi za tehnologije prepoznavanja lica.[11]

Konvolucijska neuronska mreža (CNN): Arhitektura duboke neuronske mreže koja se široko primjenjuje na obradu slike i karakteriziraju je konvolucijski slojevi koji pomiču prozore preko ulaza s čvorovima koji dijele težine, apstrahirajući (obično slikovni) ulaz u mape značajki. [11]

Rekurentna neuronska mreža (RNN) složenija je vrsta neuronske mreže. Rekurentne neuronske mreže uzimaju izlazne podatke čvora za obradu i prenose informacije natrag u mrežu. To rezultira teoretskim „učenjem“ i poboljšanjem mreže. Svaki čvor pohranjuje povijesne procese, a ti se povijesni procesi ponovno koriste u budućnosti tijekom obrade. Ovo postaje posebno kritično za mreže u kojima je predviđanje netočno. Sustav će pokušati saznati zašto je došlo do ispravnog ishoda i prilagoditi se u skladu s tim. Ova vrsta neuronske mreže često se koristi u aplikacijama za pretvaranje teksta u govor.[11]

Višeslojni perceptron neuronska mreža (MLP) je neuronska mreža sposobna rukovati i linearno odvojivim i nelinearno odvojivim podacima. Pripada klasi neuronskih mreža poznatih kao feed-forward neuronske mreže koje povezuju neurone u jednom sloju sa sljedećim slojem, na način prema naprijed, bez ikakvih petlji. [5]

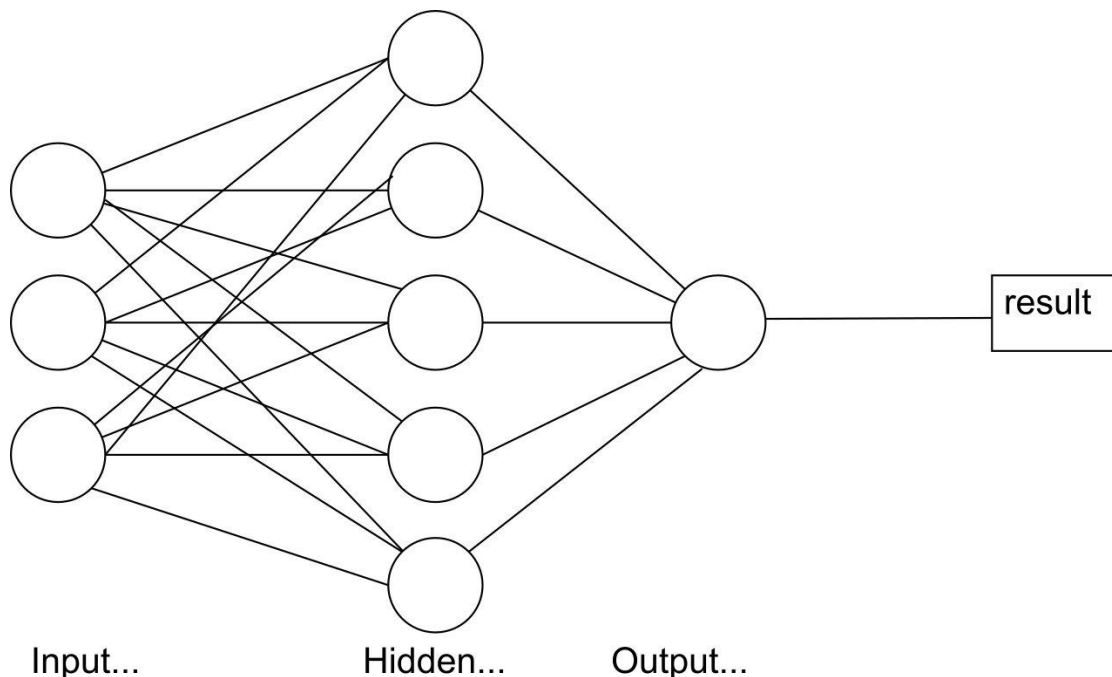
MLP je umjetna neuronska mreža i stoga se sastoji od međusobno povezanih neurona koji obrađuju podatke kroz tri ili više slojeva. Osnovna struktura MLP-a sastoji se od ulaznog sloja, jednog ili više skrivenih slojeva i izlaznog sloja, aktivacijske funkcije i skupa težina i pristranosti. Ulazni sloj je početni sloj mreže, prima ulaz u obliku brojeva. Nadalje, imamo skriveni sloj koji obrađuje informacije primljene iz ulaznog sloja. Ova obrada je u obliku izračuna. Ne postoji ograničenje broja skrivenih slojeva, iako MLP obično ima mali broj skrivenih slojeva. [5]

Konačno, posljednji sloj, izlazni sloj, odgovoran je za stvaranje rezultata. Rezultat je rezultat izračuna primijenjenih na podatke putem mreže. [5]

Još jedna karakteristika MLP-ova nalazi se u povratnom širenju, nadziranoj tehnici učenja za treniranje neuronske mreže. Pojednostavljeno rečeno, širenje unatrag je način finog podešavanja težina u neuronskoj mreži širenjem pogreške iz izlaza natrag u mrežu. Ovo poboljšava izvedbu mreže uz istovremeno smanjenje pogrešaka u izlazu. [5]

Na kraju, zbog svoje jednostavnosti, MLP-ovi će obično zahtijevati kratko vrijeme obuke kako bi naučili prikaze u podacima i proizveli izlaz. [5]

MLP se obično koriste za podatke koji nisu linearno odvojivi, kao što je regresijska analiza. Alternativno, zbog svoje jednostavnosti, najprikladniji su za složene zadatke klasifikacije i prediktivno modeliranje. Dodatno, MLP-ovi su korišteni za strojno prevođenje, vremensku prognozu, otkrivanje prijevара, predviđanje tržišta dionica i predviđanje kreditnog rejtinga.[5]



Slika 2. Struktura MLP-a

Izvor: <https://www.baeldung.com/cs/mlp-vs-dnn> (3.6.2023.)

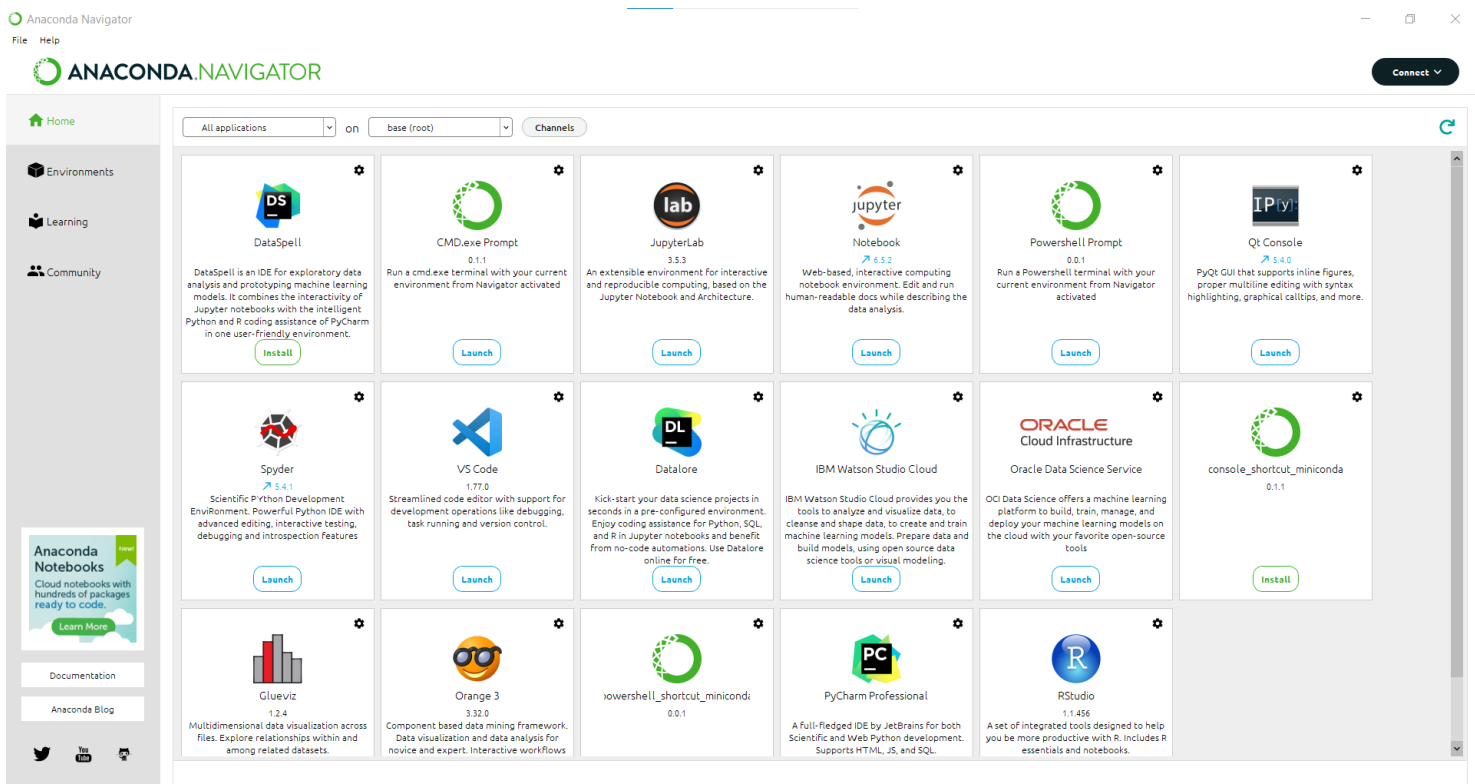
3.3. Anaconda

Anaconda je distribucija otvorenog izvornog koda programskih jezika Python i R za podatkovnu znanost koja ima za cilj pojednostaviti upravljanje paketima i njihovu implementaciju. Verzijama paketa u Anacondi upravlja sustav za upravljanje paketima, conda, koji analizira trenutno okruženje prije izvođenja instalacije kako bi se izbjeglo ometanje drugih okvira i paketa, a može se koristiti za pisanje vlastitih paketa i održavanja.

Anaconda distribucija dolazi s više od 250 automatski instaliranih paketa. Preko 7500 dodatnih open-source paketa može se instalirati iz PyPI-a, kao i paket conda i upravitelj virtualnog okruženja.[3]

Također, uključuje GUI (grafičko korisničko sučelje), Anaconda Navigator, kao grafičku alternativu sučelju naredbenog retka. Anaconda Navigator uključen je u distribuciju Anaconda i omogućuje korisnicima pokretanje aplikacija i upravljanje conda paketima, okruženjima i kanalima bez korištenja naredbi naredbenog retka. Navigator može tražiti pakete, instalirati ih u okruženje, pokretati pakete i ažurirati ih.

Anaconda olakšava povezivanje s nekoliko različitih znanstvenih paketa, paketa Machine Learning i Data Science.[3].



Slika 3. Prikaz početne strane anaconda navigator.

Izvor: Autor

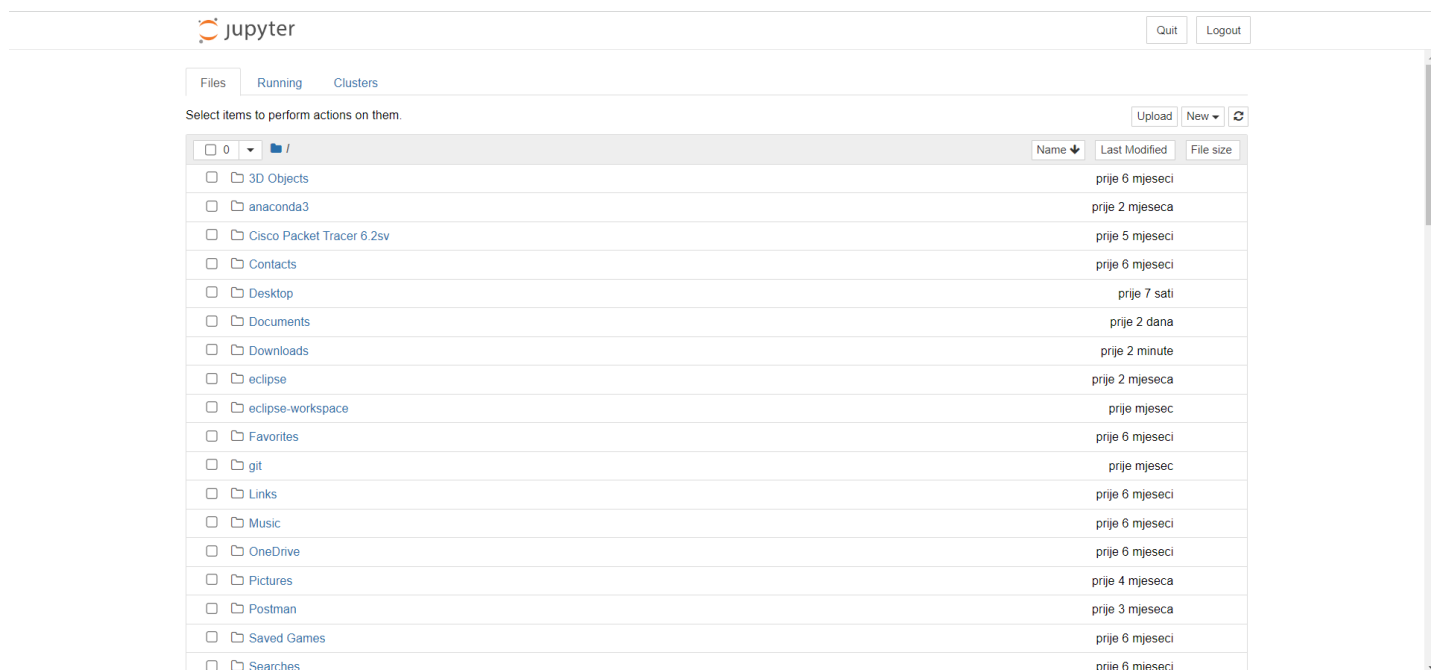
3.3.1. Jupyter notebook

Jupyter Notebook web je aplikacija otvorenog koda koja znanstvenicima podataka omogućuje stvaranje i dijeljenje dokumenata koji integriraju kod, jednadžbe, računalni izlaz, vizualizacije i druge multimedijске resurse, zajedno s tekstom objašnjenja u jednom dokumentu. Može se koristiti za sve vrste zadataka o podacima, uključujući čišćenje i transformaciju podataka, numeričku simulaciju, istraživačku analizu podataka, vizualizaciju podataka, statističko modeliranje, strojno učenje, duboko učenje i još mnogo toga.[6]

Jupyter bilježnica ima dvije komponente. Prvo, unosi se programski kod ili tekst u pravokutne „ćelije“ na početnoj web stranici. Preglednik zatim prosljeđuje kod pozadinskom „kernelu“ koji pokreće kod i vraća rezultate. Stvorene su mnoge Jupyter jezgre koje podržavaju desetine programskih jezika. Kerneli se ne moraju nalaziti na računalu za podatke. Može se pokrenuti bez pristupa mreži izravno na vlastitom računalu i obavljati svoj posao lokalno.[6]

Jupyter Notebook također omogućuje uključivanje grafičkih prikaza, kao što su grafikoni i dijagrami, te interaktivnih elemenata poput widgeta koji korisniku omogućuju kontrolu nad izvršavanjem koda ili vizualizacijom rezultata. Može izvesti Notebook u različitim formatima datoteka, poput PDF-a ili HTML-a, kako bi ga jednostavno podijelili s drugima.[8]

Jupyter Notebooks su spin-off projekti iz projekta IPython koji je prije imao sam projekt IPython Notebook. Naziv Jupyter dolazi od osnovnih podržanih programskih jezika koje podržava: Julia, Python i R. Jupyter se isporučuje s IPython kernelom koji vam omogućuje pisanje programa u Pythonu.[7]



Slika 4. Prikaz sučelja jupyter notebook-a.

Izvor: Autor

3.4. Python

Python je interpretirani, objektno orijentirani programski jezik visoke razine s dinamičkom semantikom koju je razvio Guido van Rossum. Izvorno je objavljen 1991. Dizajniran da bude jednostavan i zabavan, naziv „Python" je znak britanske komičarske grupe Monty Python. Python ima reputaciju jezika prilagođenog početnicima, zamjenjujući Javu kao najčešće korišteni uvodni jezik jer se nosi s velikim dijelom složenosti za korisnika, omogućujući početnicima da se usredotoče na potpuno shvaćanje koncepata programiranja, a ne na sitne detalje. [3]

Python se koristi za web razvoj na strani poslužitelja, razvoj softvera, matematiku i skriptiranje sustava, a popularan je za brzi razvoj aplikacija i kao skriptni ili ljepljivi jezik za povezivanje postojećih komponenti zbog svojih ugrađenih struktura podataka visoke razine, dinamičko tipkanje i dinamičko uvezivanje. Troškovi održavanja programa smanjeni su s Pythonom zbog sintakse koja se lako uči i naglaska na čitljivosti. Dodatno, Pythonova podrška za module i pakete olakšava modularne programe i ponovnu upotrebu koda. Python je jezik zajednice otvorenog koda pa brojni neovisni programeri neprestano izgrađuju biblioteke i funkcionalnosti za njega. [3]

U profesionalnom smislu, Python je izvrstan za backend web razvoj, analizu podataka, umjetnu inteligenciju i znanstveno računalstvo. Programeri također koriste Python za izradu alata za produktivnost, izradu igara i aplikacija za stolna računala. Python, dinamički tipizirani jezik, posebno je fleksibilan, eliminira čvrsta pravila za izgradnju značajki i nudi veću fleksibilnost u rješavanju problema s različitim metodama. Također, omogućuje korisnicima da prevedu i pokrenu programe sve do problematičnog područja jer koristi provjeru tipa u vrijeme izvođenja, umjesto provjere u vrijeme prevođenja i jednostavan je programski jezik. [3]

4.KORIŠTENE PYTHON BIBLIOTEKE

U ovom poglavlju objašnjene su biblioteke koje su potrebne i korištene kako bi aplikacija ispravno radila i pružila potrebne funkcionalnosti. Bez tih biblioteka aplikacija ne bi mogla biti izrađena i ne bi bila funkcionalna.

```
1 import pandas as pd
2 import numpy as np
3 import tkinter as tk
4 import matplotlib.pyplot as plt
5 from sklearn.model_selection import train_test_split
6 from sklearn.neural_network import MLPRegressor
7 from sklearn.preprocessing import StandardScaler
8 from sklearn.metrics import mean_squared_error, r2_score
9 from datetime import datetime
10 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
11 from PyQt5.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QWidget, QScrollArea, QTableWidgetItem,
12 QTableWidgetItem, QLabel, QPushButton, QTextEdit, QMessageBox, QFileDialog, QPlainTextEdit
13 from tkinter import ttk
14 from tkinter import Tk
15 from PIL import ImageTk, Image
16 from PyQt5.QtGui import QImage, QPixmap
17 from PyQt5.QtCore import Qt
18 from PyQt5.QtGui import QTextCursor, QTextCharFormat, QColor, QFont
19
```

Slika 5. Prikaz korištenih biblioteka u aplikaciji

Izvor: Autor

4.1 Pandas

Python Pandas se definira kao biblioteka otvorenog koda i koristi se u znanosti o podacima, podacima o analizi i aktivnosti strojnog učenja. Nudi niz struktura podataka i operacija za rad s vremenskim serijama i numeričkim podacima. Ova je biblioteka razvijena na temelju biblioteke NumPy koja podržava višedimenzionalne nizove. Pandas je jedan od najčešće korištenih paketa za obradu podataka. [14]

Analiza podataka zahtijeva puno obrade, kao što je restrukturiranje, čišćenje, spajanje itd. [15]

4.1.1. NumPy

NumPy, kratica je za Numerical Python, to je biblioteka koja se sastoji od višedimenzionalnih objekata nizova i zbirke rutina za obradu tih nizova. Koristeći NumPy, mogu se izvoditi matematičke i logičke operacije nad nizovima. [16]

4.1.2. Matplotlib

Matplotlib je osnovna biblioteka za crtanje programskog jezika Python. Među Python paketima za vizualizaciju, on je najčešće korišten. Iznimno je brz u raznim operacijama. Osim toga, može izvesti vizualizacije u sve popularne slikovne formate. [17] Može izraditi linijske grafikone, raspršene dijagrame, histograme, stupčaste dijagrame, dijagrame pogrešaka, tortne dijagrame, okvirne dijagrame i mnoge druge stilove vizualizacije. Uz to, 3D grafikoni su također mogući uz Matplotlib. [17]

4.1.3. Sklearn

Scikit-learn (Sklearn) je najkorisnija i najsnažnija biblioteka za strojno učenje u Pythonu. Pruža izbor učinkovitih alata za strojno učenje i statističko modeliranje uključujući klasifikaciju, regresiju, grupiranje i smanjenje dimenzionalnosti putem konzistentnog sučelja u Pythonu. Ova biblioteka, koja je uglavnom napisana u Pythonu, izgrađena je na NumPy, SciPy i Matplotlib . [18]

4.1.4. Tkinter

Tkinter je ugrađeni Python modul koji se koristi za izradu GUI (Graphical User Interface) aplikacija. To je jedan od najčešće korištenih modula za izradu GUI aplikacija u Pythonu jer je s njim jednostavno i lako raditi. Daje objektno orijentirano sučelje za Tk GUI toolkit. [19]

Tkinter se najviše koristi. Postoje još neke druge popularne biblioteke za izradu GUI aplikacije, a to su: Kivy, Python Qt, wxPython. [19]

4.1.5. DateTime

Python Datetime modul nudi klase za rad s datumom i vremenom. Ove klase pružaju brojne funkcije za rad s datumima, vremenima i vremenskim intervalima. [20]

Modul DateTime je kategoriziran u 6 glavnih klasa, a to su: date, time, datetime, timedelta, tzinfo, timezone. [20]

4.1.6. Pillow

Modul Python Pillow izgrađen je na temelju PIL-a (Python Image Library). To su osnovni moduli za obradu slika u Pythonu i možemo učiniti bilo što na digitalnim slikama pomoću tog modula. [21]

4.1.7. PyQt5

PyQt je skup alata za widgete grafičkog korisničkog sučelja. Radi kao Python sučelje za Qt, jednu od najistaknutijih i najpoznatijih višeplatformskih GUI biblioteka. [21]

PyQt5 se temelji na Qt verziji 5 i sadrži klase koje pokrivaju grafička korisnička sučelja kao i mrežnu komunikaciju, multimediju, rukovanje XML-om, regularne izraze, SQL baze podataka, multimediju, pregledavanje weba i druge tehnologije dostupne u Qt-u. Biblioteka PyQt5 je Python paket najviše razine koji može implementirati više od tisuću ovih Qt klasa u skupu Python modula. [21]

4.1.8. Requests

Biblioteka requests koristi se za izradu HTTP zahtjeva.[22]

4.1.9. Beautiful Soup

Beautiful Soup je Python biblioteka za izvlačenje podataka iz HTML i XML datoteka.[22]

4.2.0. Csv

Modul csv implementira klase za čitanje i pisanje tabličnih podataka u CSV formatu. [24]

5.SKUPLJANJE PODATAKA PUTEM WEB SCRAPING-A ZA APLIKACIJU

U ovom poglavlju je prikazano i objašnjeno kako autor prikuplja podatke za 3 meteo-stanice koje je odabrao, a to su: Čakovec, Varaždin i Koprivnica.

Autor je proveo proces skupljanja podataka za mjesec travanj 2023. kako bi prikupio vremenske informacije putem web scraping-a s web stranice „Google Weather“. Skupljao je sljedeće podatke: dan, mjesto, trenutna temperatura, oborine, vlaga, vjetar, trenutno vrijeme, najveća temperatura, najmanja temperatura.

Autor je putem procesa web scraping-a automatski dohvaćao podatke s web stranice „Google Weather“ za svaki dan u mjesecu travnju 2023. Nakon prikupljanja podataka za određeno mjesto, svaki set podataka je spremljen u zasebnu excel datoteku. Ovo je omogućilo autoru da kasnije koristi prikupljene podatke u aplikaciji za analizu i prikazivanje vremenskih informacija korisnicima.

5.1. Automatsko dohvaćanje podataka

U ovom dijelu rada autor je koristio Python kod u jupyter notebook- kako bi automatski dohvaćao podatke o vremenskim uvjetima za Čakovec putem web scraping-a s Google pretraživača.

Autor je koristio biblioteke requests i BeautifulSoup za slanje HTTP zahtjeva na Google pretraživač i analizu HTML sadržaja dobivenog kao odgovor.

Kod započinje definiranjem URL-a koji vodi do web stranice koja prikazuje vremenske podatke za Čakovec. Zatim se šalje HTTP zahtjev na taj URL kako bi se dobio HTML sadržaj stranice. Nakon toga, autor je analizirao HTML sadržaj kako bi pronašao i izdvojio relevantne informacije o vremenskim uvjetima.

User-Agent zaglavlje omogućuje web poslužitelju da prepozna koja vrsta web preglednika šalje zahtjev. Koristi se u HTTP zahtjevu kako bi se identificirao kao preglednik. Svaki preglednik i korisnik ima svoj User-Agent.

Pomoću identificiranih HTML elemenata koji sadrže podatke o danu, mjestu, trenutnoj temperaturi, oborinama, vlazi, vjetru, trenutnom vremenu, najvećoj i najmanjoj temperaturi, autor je dohvaćao potrebne podatke i prikazao je dohvaćene podatke u konzoli koristeći print funkciju.

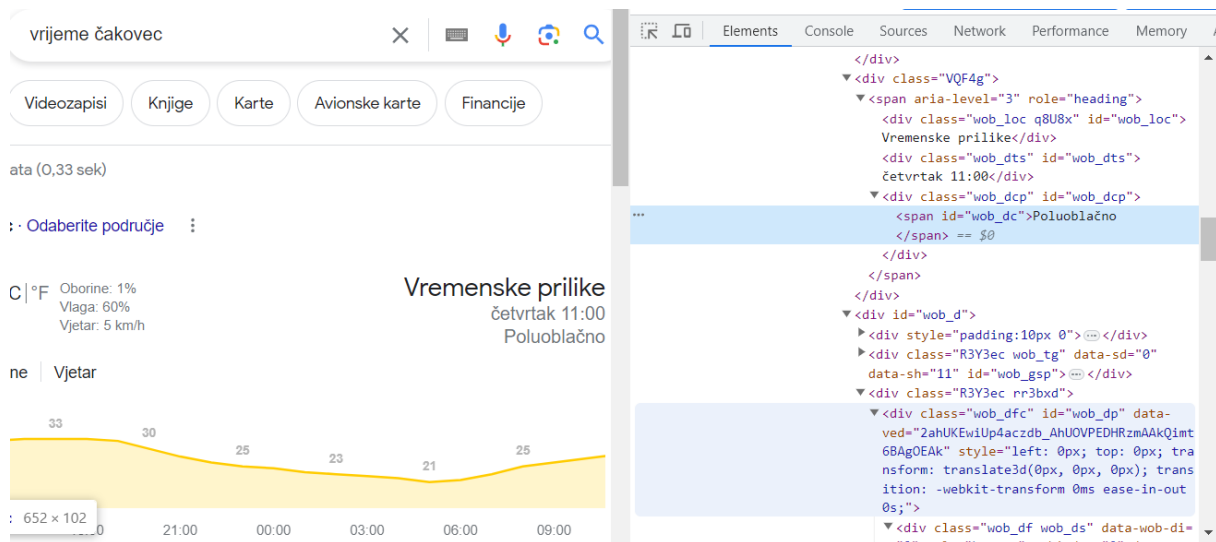
```

1 import requests
2 from bs4 import BeautifulSoup
3 import csv
4 url = "https://www.google.com/search?q=vrijeme+Čakovec"
5 headers = {
6     "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/111.0.0.0 Safari/
7 }
8 page = requests.get(url, headers=headers)
9 soup = BeautifulSoup(page.content, "html.parser")
10
11 dan = soup.find("div", attrs={"id": "wob_dts"}).text
12 mjesto=soup.find("span", attrs={"class": "BBwThe"}).text
13 temp = soup.find("span", attrs={"id": "wob_tm"}).text
14 oborine=soup.find("span", attrs={"id": "wob_pp"}).text
15 vlaga=soup.find("span", attrs={"id": "wob_hm"}).text
16 vjetar=soup.find("span", attrs={"id": "wob_ws"}).text
17 trenut=soup.find("span", attrs={"id": "wob_dc"}).text
18 max_temp = soup.find("div", attrs={"class": "wob_df"}).find_next("span").text
19 min_temp=soup.find("div", attrs={"class": "QrNVmd ZXCv8e"}).find_next('span', attrs={'class': 'wob_t'}).text
20
21 oborine = oborine.split()[0].strip('%')
22 vlaga = vlaga.split()[0].strip('%')
23 vjetar = vjetar.split()[0].strip()
24 print(f"Dan: {dan}")
25 print(f"Mjesto: {mjesto}")
26 print(f"Temperatura: {temp}")
27 print(f"Oborine: {oborine}")
28 print(f"Vlaga: {vlaga}")
29 print(f"Vjetar: {vjetar}")
30 print(f"Trenutno vrijeme: {trenut}")
31 print(f"Najveća temperatura: {max_temp}")
32 print(f"Najmanja temperatura: {min_temp}")

```

Slika 6. Prikaz automatskog dohvaćanja podataka.

Izvor: Autor



Slika 7. Identificirani HTML element.

Izvor: Autor

```
Dan: četvrtak 13:00
Mjesto: Čakovec
Temperatura: 31
Oborine: 0
Vlaga: 58
Vjetar: 6
Trenutno vrijeme: Poluoblačno
Najveća temperatura: 33
Najmanja temperatura: 21
```

Slika 8. Prikaz dohvaćenih podataka u konzoli.

Izvor: Autor

5.1.1. Spremanje dohvaćenih podataka.

U ovom dijelu rada autor koristi kod kako bi otvorio csv, excel datoteku 'Cakovec.csv', dodao podatke o vremenskim uvjetima za određeno mjesto (Čakovec).

U prvom retku se pišu nazivi stupaca u datoteci, a u drugom se retku pišu stvarni podaci koji su prikupljeni tijekom web scraping-a koji se spremaju unutar datoteke. Ovaj korak je važan jer omogućuje da se prikupljeni podatci spremaju u strukturirani format (csv) koji se može koristiti u daljnjoj obradi.

```
1 with open('Cakovec.csv', mode='a', newline='') as file:
2     writer = csv.writer(file, delimiter=';')
3     writer.writerow(['Dan', 'Mjesto', 'Trenutačna Temperatura', 'Oborine', 'Vlaga', 'Vjetar', 'Trenutnovrijeme',
4                     'Najveća temperatura', 'Najmanja temperatura'])
5     writer.writerow([dan, mjesto, temp, oborine, vlaga, vjetar, trenut, max_temp, min_temp])
6
```

Slika 9. Prikaz koda za unos podataka u excel.

Izvor: Autor

	A	B	C	D	E	F	G	H	I	J
1	Dan	Mjesto	Trenutačn:	Oborine	Vlaga	Vjetar	Trenutnov	Najvećate	Najmanjate	temperatura
2	srijeda 16:	Čakovec	8	0	36	16	Poluoblač	8	-2	
3	četvrtak 2:	Čakovec	6	0	55	6	Poluoblač	11	-1	
4	petak 23:0	Čakovec	6	73	73	6	Oblačno	12	3	
5	subota 13:	Čakovec	7	90	85	4	Kiša	8	4	

Slika 10. Prikaz excel-a s podacima.

Izvor: Autor

Ovaj proces skupljanja podataka putem web scraping-a bio je neophodan za daljnji rad aplikacije jer su ti podaci korišteni za daljnju analizu i prikazivanje vremenskih informacija u aplikaciji.

6.FUNKCIONALNOST APLIKACIJE

U ovom poglavlju objašnjene su i prikazane sve glavne značajke aplikacije, a to su : Treniranje i testiranje, statistika, vrijeme, vremenska prognoza, treniranje-testiranje-pojedinačno.

Aplikacija ima jedno glavno sučelje u kojoj su prikazane sve glavne značajke i 5 sučelja koje su dio glavnih značajki, i gumbове za svaku značajku. Pritiskom na određenu značajku, otvara se novo sučelje s tom značajkom koju smo odabrali. Kako bismo se uvijek vratili na glavno sučelje, potrebno je minimizirati trenutno sučelje ili zatvoriti.

Autor je dodao da svako sučelje ima isti stil koji se primjenjuje na sve gumbове kako bi se postigao konzistentan izgled i poboljšala interakcija s korisnikom u sučeljima, osim u sučelju vremenska prognoza, u kojem autor koristi Tkinter sučelje.

6.1. Glavno sučelje

U ovom dijelu rada, ovaj kod stvara glavno sučelje s pozadinom slike i gumbima koji su povezani s odgovarajućim funkcijama prikaza drugih sučelja. Svaki gumb ima svoju funkcionalnost koja će se izvršiti kada se klikne na njega.

Prvo se postavlja naslov prozora i veličina prozora, a zatim se učitava slika s lokacije koju je autor odabrao.

Nakon toga se stvaraju gumbi, svaki s odgovarajućim natpisom, stilom i postavljanjem veličine i položaja.

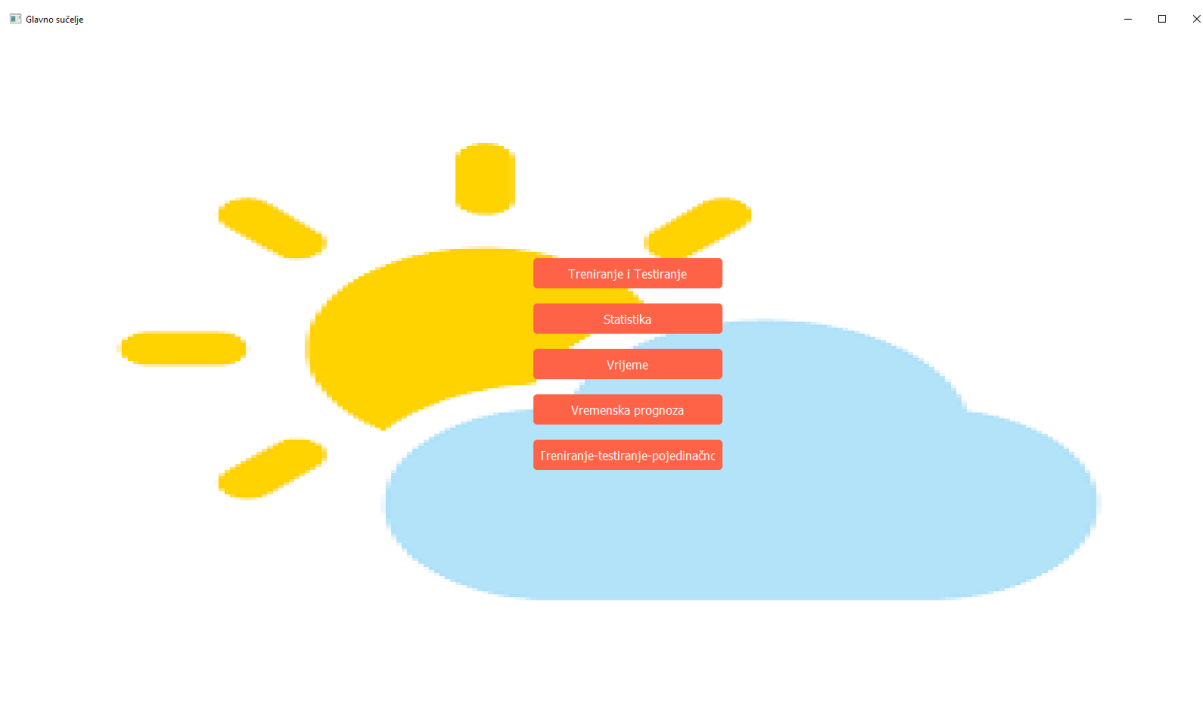
```

192 class MainWindow(QMainWindow):
193     def __init__(self):
194         super().__init__()
195         self.setWindowTitle("Glavno sučelje")
196         self.setGeometry(100, 100, 1600, 900)
197         image = QImage('C:/Users/fbara/Desktop/images.png')
198         image = image.scaled(1600, 900)
199         label = QLabel(self)
200         label.setPixmap(QPixmap.fromImage(image))
201         label.setGeometry(0, 0, 1600, 900)
202         label.setAlignment(Qt.AlignCenter)
203         image_opacity = 0.7
204         button_styles = '''
205         QPushButton {
206             background-color: #fff6347; /* Promijenite boju ovdje */
207             color: white;
208             border: none;
209             border-radius: 5px;
210             padding: 10px;
211             font-size: 16px;
212         }
213         QPushButton:hover {
214             background-color: #fff4d26; /* Promijenite boju ovdje */
215         }
216         '''
217         self.button4 = QPushButton("Treniranje i Testiranje", self)
218         self.button4.setGeometry(700, 300, 250, 40)
219         self.button4.setStyleSheet(button_styles)
220         self.button4.clicked.connect(self.show_interface5)
221         self.button1 = QPushButton("Statistika", self)
222         self.button1.setGeometry(700, 360, 250, 40)
223         self.button1.setStyleSheet(button_styles)
224         self.button1.clicked.connect(self.show_interface1)
225         self.button2 = QPushButton("Vrijeme", self)
226         self.button2.setGeometry(700, 420, 250, 40)
227         self.button2.setStyleSheet(button_styles)
228         self.button2.clicked.connect(self.show_interface2)
229         self.button3 = QPushButton("Vremenska prognoza", self)
230         self.button3.setGeometry(700, 480, 250, 40)
231         self.button3.setStyleSheet(button_styles)
232         self.button3.clicked.connect(self.show_interface3)
233         self.button = QPushButton("Treniranje-testiranje-pojedinačno", self)
234         self.button.setGeometry(700, 540, 250, 40)
235         self.button.setStyleSheet(button_styles)
236         self.button.clicked.connect(self.show_interface4)
237         self.current_interface = None

```

Slika 11. Prikaz koda za glavno sučelje.

Izvor: Autor



Slika 12. Prikaz glavnog sučelja.

Izvor: Autor

6.2. Treniranje i testiranje sučelje

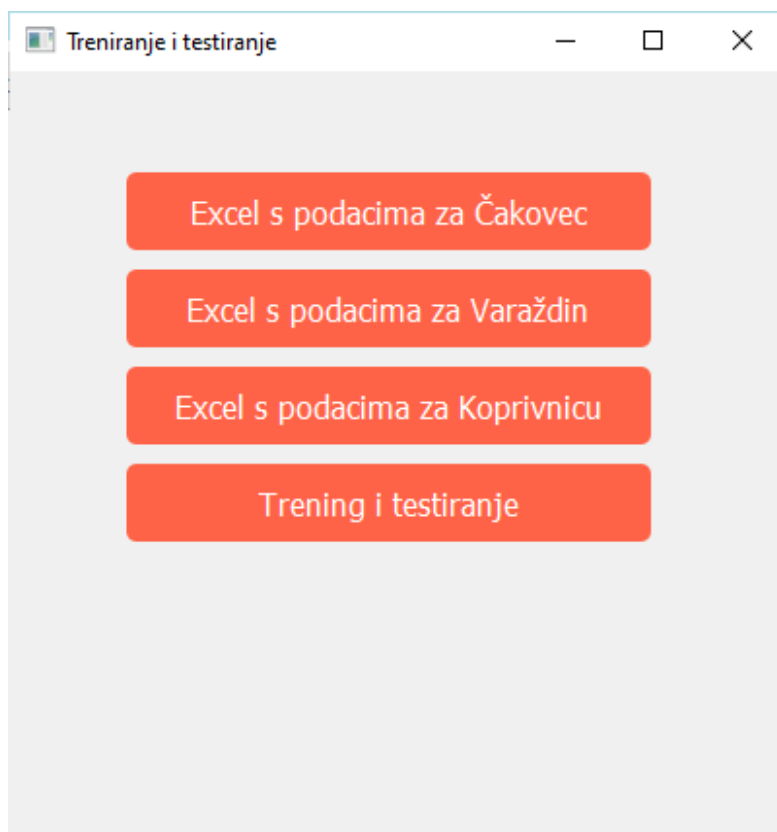
Kad je korisnik u glavnom sučelju kliknuo na gumb „Treniranje i testiranje“, otvorit će mu se sučelje koje je implementirano putem klase `TreniranjeTesitranjeApp`. Ovo sučelje je nazvano Treniranje i testiranje.

Na sučelju se nalaze četiri gumba s odgovarajućim nazivima. Prvi gumb, naziva „Excel s podacima za Čakovec“, omogućuje korisniku da odabere Excel datoteku koja sadrži podatke za grad Čakovec.

Svaki gumb ima svoje dimenzije, stil i poziciju na sučelju te je povezan s odgovarajućom funkcijom koja će se izvršiti kada se gumb klikne.

Slično tome, drugi gumb, „Excel s podacima za Varaždin“, omogućuje odabir Excel datoteke s podacima za grad Varaždin, dok treći gumb, „Excel s podacima za Koprivnicu“, omogućuje odabir Excel datoteke s podacima za grad Koprivnicu.

Četvrti gumb na sučelju, naziva „Trening i testiranje“, omogućuje pokretanje procesa treniranja i testiranja modela na temelju odabranih podataka. Kada se gumb klikne, izvršit će se funkcija `predict_weather`, koja će inicirati proces treniranja i testiranja modela na temelju odabranih Excel datoteka.



Slika 13. Prikaz sučelja Treniranje i testiranje.

Izvor: Autor

```
23 class TreniranjeTestiranjeApp(QMainWindow):
24     def __init__(self, parent=None):
25         super().__init__()
26
27         self.setWindowTitle("Treniranje i testiranje")
28         self.setGeometry(100, 100, 300, 300)
29
30
31
32         button_styles = '''
33         QPushButton {
34             background-color: #ff6347;
35             color: white;
36             border: none;
37             border-radius: 5px;
38             padding: 10px;
39             font-size: 16px;
40         }
41         QPushButton:hover {
42             background-color: #ff4d26;
43         }
44         ...
45
46         self.button1 = QPushButton("Excel s podacima za Čakovec", self)
47         self.button1.setGeometry(60, 52, 270, 40)
48         self.button1.setStyleSheet(button_styles)
49         self.button1.clicked.connect(self.select_df)
50
51         self.button2 = QPushButton("Excel s podacima za Varaždin", self)
52         self.button2.setGeometry(60, 102, 270, 40)
53         self.button2.setStyleSheet(button_styles)
54         self.button2.clicked.connect(self.select_df_varazdin)
55
56         self.button3 = QPushButton("Excel s podacima za Koprivnicu", self)
57         self.button3.setGeometry(60, 152, 270, 40)
58         self.button3.setStyleSheet(button_styles)
59         self.button3.clicked.connect(self.select_df_koprivnica)
60
61         self.button4 = QPushButton("Trening i testiranje", self)
62         self.button4.setGeometry(60, 202, 270, 40)
63         self.button4.setStyleSheet(button_styles)
64         self.button4.clicked.connect(self.predict_weather)
```

Slika 14. Prikaz koda sučelja Treniranje i testiranje.

Izvor: Autor

6.2.1. Funkcije treniranje i testiranje sučelja.

Kad je korisnik kliknuo na gumb „Excel s podacima za Čakovec“, otvara mu se dijaloški prozor za odabir excel datoteke. Za svaki gumb Excel s podacima implementira se funkcija `select_df`.

Nakon što je korisnik odabrao excel datoteku, učitavaju se podatci iz datoteka i implementira se funkcija `load_df`.

Učitani podaci se pohranjuju u globalne varijable kako bi bili dostupni za daljnje obrade.

```
42 def select_df(self):
43     file_dialog = QFileDialog()
44     file_path, _ = file_dialog.getOpenFileName(self, "Odaberi excel čakovec datoteku", "", "CSV Files (*.csv)")
45
46     if file_path:
47         self.load_df(file_path)
48
49 def load_df(self, file_path):
50     global df
51     df = pd.read_csv(file_path, encoding='cp1250', delimiter=";")
52
53 def select_df_varazdin(self):
54     file_dialog = QFileDialog()
55     file_path, _ = file_dialog.getOpenFileName(self, "Odaberi excel varaždin datoteku", "", "CSV Files (*.csv)")
56
57     if file_path:
58         self.load_df_varazdin(file_path)
59
60 def load_df_varazdin(self, file_path):
61     global df_varazdin
62     df_varazdin = pd.read_csv(file_path, encoding='cp1250', delimiter=";")
63
64 def select_df_koprivnica(self):
65     file_dialog = QFileDialog()
66     file_path, _ = file_dialog.getOpenFileName(self, "Odaberi excel koprivnica datoteku", "", "CSV Files (*.csv)")
67
68     if file_path:
69         self.load_df_koprivnica(file_path)
70
71 def load_df_koprivnica(self, file_path):
72     global df_koprivnica
73     df_koprivnica = pd.read_csv(file_path, encoding='cp1250', delimiter=";")
74
```

Slika 15. Prikaz koda funkcija za odabir i učitavanje podataka.

Izvor: Autor

Nakon što su odabrane excel datoteke, i kad korisnik klikne na gumb „Treniranje i testiranje“, poziva se funkcija `predict_weather` koja je iznimno važna i ona obavlja ključne zadatke.

Provjerava postoje li podaci iz excel datoteka i prikazuje upozorenje ako neki od excel-a nije odabran. Ako jedan od njih nije odabran, funkcija se prekida, ako su svi dostupni, spaja ih se u jedan DataFrame. Ovo je ključno jer se bez podataka ne može krenuti na daljnju obradu.

Pripremaju se podaci za treniranje modela. Stupac „Mjesto” se ažurira kako bi se uskladio s jedinstvenim nazivima za mjesta i dodaje se stupac „Datum” u DataFrame koji sadrži raspon datuma.

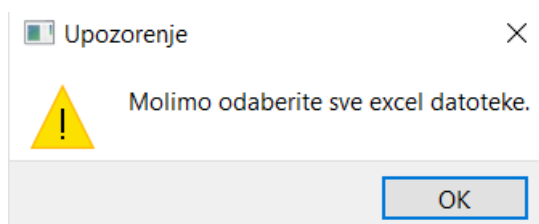
```

75 def predict_weather(self):
76     global df, df_varazdin, df_koprivnica, new_df_2024,r2,mse,new_df,merged_df
77
78     if df.empty or df_varazdin.empty or df_koprivnica.empty:
79         QMessageBox.warning(self, "Upozorenje", "Molimo odaberite sve excel datoteke.")
80         return
81     df['Mjesto']
82     df['Dan']
83     for index, row in df.iterrows():
84         if row['Mjesto'] == 'Čakovec':
85             df.at[index, 'Mjesto'] = 'Čakovec'
86         elif row['Mjesto'] == 'Varaždin':
87             df.at[index, 'Mjesto'] = 'Varaždin'
88         elif row['Mjesto'] == 'Koprivnica':
89             df.at[index, 'Mjesto'] = 'Koprivnica'
90
91
92     df = pd.concat([df, df_varazdin, df_koprivnica], ignore_index=True)
93

```

Slika 16. Prikaz koda za provjeravanje i pripremu podataka.

Izvor: Autor



Slika 17. Prikaz poruke s greškom

Izvor: Autor

Vrši se zamjena tekstualnih vrijednosti u stupcu „Trenutno vrijeme” numeričkim vrijednostima od 0 do 8. Svaka numerička vrijednost predstavlja određeno stanje vremena. Ovo je ključno jer modeli strojnog učenja ne mogu izravno raditi s tekstualnim podacima, oni zahtijevaju numeričke vrijednosti kako bi mogli trenirati i testirati svoje modele. Stoga je autor primijenio postupak zamjene koristeći tehniku nazvanu 'one-hot encoding'. Nakon zamjene, model može uspješno trenirati i testirati koristeći numeričke vrijednosti i donositi predviđanja na temelju tih vrijednosti.

```

95 df['Trenutnovrijeme'] = df['Trenutnovrijeme'].replace(['Kiša', 'Oblačno', 'Poluoblačno', 'Slabi pljuskovi uz kišu',
96 'Uglavnom oblačno', 'Uglavnom sunčano', 'Vedro',
97 'Vedro s povremenom naoblakom'], [0,1,2,3,4,5,6,7,8])
98

```

Slika 18. Prikaz koda za zamjenu tekstualnih vrijednosti.

Izvor: Autor

Podaci se dijele na skupove za treniranje i testiranje te se standardiziraju pomoću StandardScaler. Dijele se na ulazne varijable (X) i ciljne varijable (y). Autor je odabrao sljedeće podatke za ulazne i ciljne varijable: Trenutačna temperatura, Oborine, Vlaga, Vjetar, Trenutno vrijeme, Najveća temperatura, Najmanja temperatura.

Kreira i trenira model MLPRegressor (Multi-Layer Perceptron Regressor) iz biblioteke sklearn.neural_network. Model se trenira na podacima za treniranje (X_train i y_train). Autor je podijelio podatke na skupove za treniranje i testiranje primjenom omjera 70:30. To znači da je 70% podataka dodijeljeno skupu za treniranje, dok je preostalih 30% podataka dodijeljeno skupu za testiranje.

```
99 x = ['Oborine', 'Vlaga', 'Vjetar', 'Trenutnovrijeme', 'Najvećatemperatura', 'Najmanjatemperatura']
100 y = ['Oborine', 'Vlaga', 'Vjetar', 'Trenutnovrijeme', 'Najvećatemperatura', 'Najmanjatemperatura']
101
102
103 X_train, X_test, y_train, y_test = train_test_split(df[X], df[y], test_size=0.3, random_state=43)
104
105
106 scaler = StandardScaler()
107 X_train = scaler.fit_transform(X_train)
108 X_test = scaler.transform(X_test)
109
```

Slika 19. Prikaz koda podijele podataka na skupove i kreiranje modela.

Izvor: Autor

Model se koristi za predviđanje izlaznih vrijednosti na temelju podataka za testiranje. Izračunava se srednja kvadratna greška (mse) i r2 koeficijent determinacije (r2_score) između stvarnih vrijednosti i predviđenih vrijednosti, kako bi se evaluirala točnost modela. R2 vrijednost se kreće od 0 do 1, što je vrijednost bliže 1 ukazuje na savršenu usklađenost modela s podacima, MSE je uvijek pozitivan broj, a manje vrijednosti ukazuju na bolju preciznost modela.

Rezultati koje je autor dobio u aplikaciji upućuju na to da je model dobro naučio podatke i ima visoku prediktivnu moć. Autor je dobio sljedeće rezultate:

srednja kvadratna greška: 1.550027301975588, koeficijent determinacije:
0.9772918039864524

```
111 mlp = MLPRegressor(hidden_layer_sizes=(100,50,100, 50), max_iter=1500, solver='adam', activation='relu', random_state=43)
112 mlp.fit(X_train, y_train)
113
114
115 y_pred = mlp.predict(X_test)
116
117
118 mse = mean_squared_error(y_test, y_pred)
119 r2 = r2_score(y_test, y_pred)
120 print('Mean Squared Error:', mse)
121 print('R^2:', r2)
122
123
124
125
126
127 new_dates = pd.date_range(start=datetime(2024, 4, 1), periods=len(df))
128 new_df = df.copy()
129
130
131 new_X = scaler.transform(new_df[X])
132 new_y = mlp.predict(new_X)
133
134 for i, param in enumerate(y):
135     new_df[param] = new_y[:, i]
136
```

Slika 20. Prikaz koda predviđenih izlaznih vrijednosti i spremanja.

Izvor: Autor

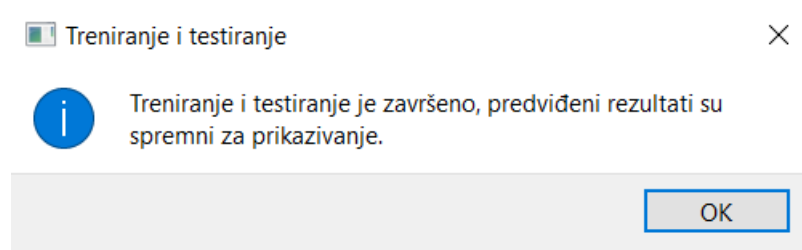
Predviđeni rezultati se spremaju u novi DataFrame, zatim se spajaju originalni DataFrame i novi, koristeći funkciju concat, nazvan je novi DataFrame s pripadajućim datumima za 2024.

Ako je sve ispravno prošlo, korisniku će se otvoriti QMessageBox s uspješnom porukom, „Treniranje i testiranje je završeno, predviđeni rezultati su spremni za prikazivanje.”

```
152 merged_df = pd.concat([df, new_df], ignore_index=True).round()
153 merged_df=merged_df.round(2)
154
155 merged_df['Trenutnovrijeme'] = merged_df['Trenutnovrijeme'].replace(trenutnovrijeme_dict)
156 merged_df['Oborine'] = merged_df['Oborine'].apply(lambda x: abs(x))
157 new_df_2024 = merged_df.iloc[len(df):].reset_index(drop=True)
158 new_dates_2024 = pd.date_range(start=pd.Timestamp(2024, 4, 1), periods=len(new_df_2024), freq='D')
159 new_df_2024['Datum'] = new_dates_2024
160 new_df_2024 = new_df_2024[y].round(2)
161 new_df_2024.insert(6, 'Datum', new_dates_2024)
162 new_df_2024['Mjesto'] = df['Mjesto']
163 new_df_2024['Dan'] = df['Dan']
164
165
166 QMessageBox.information(self, "Predviđanje vremena", "Predviđanje je završeno, rezultati su spremni za prikazivanje.")
167 pass
168
```

Slika 21. Prikaz koda za prikazivanje poruke.

Izvor: Autor



Slika 22. Prikaz uspješne poruke.

Izvor: Autor

6.2. Statistika sučelje

Kad je korisnik u glavnom sučelju kliknuo na gumb „Statistika“, otvorit će mu se sučelje koje je implementirano putem klase StatistikaApp.

Ako korisnik nije prvo odabrao sve excel datoteke i napravio treniranje i testiranje, pojavit će se poruka s greškom, „odaberite excel datoteke kako bi se prikazala statistika“.

Sučelje ima tri gumba („Prikaži R^2 i MSE“, „Prikaži statistiku za 2023“ i „Prikaži statistiku za 2024“) povezana s odgovarajućom funkcijom koja će se izvršiti kada se gumb klikne, a dodani su i stilovi za svaki gumb.

Gumb naziva „Prikaži R^2 i MSE“ omogućuje korisniku da mu se prikaže vrijednost srednje kvadratne greške i r^2 koeficijent determinacije, povezan je s funkcijom „r2_mse“.

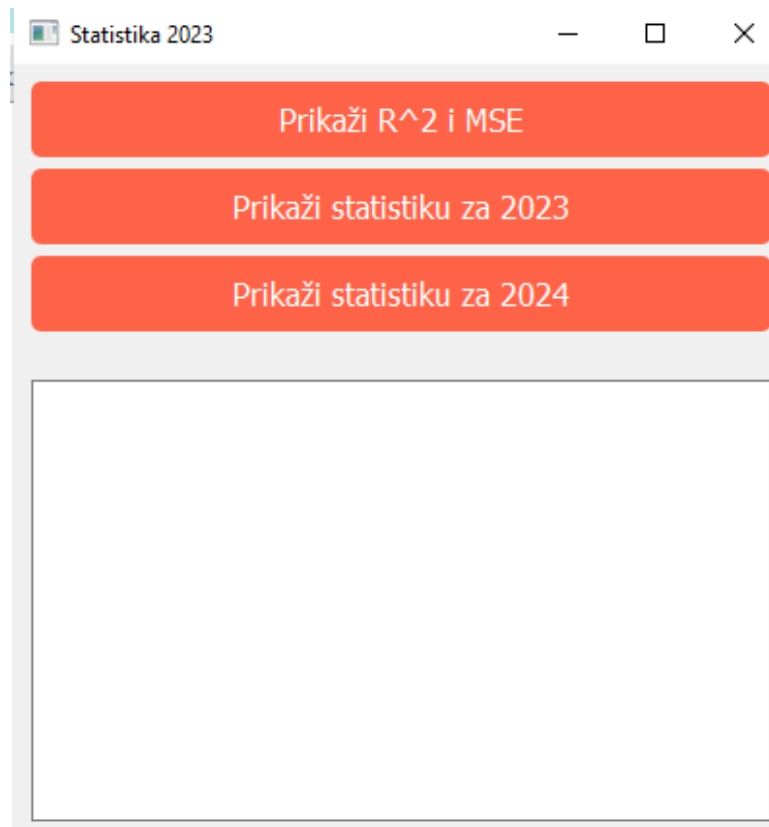
Drugi gumb „Prikaži statistiku za 2023“ prikazuje korisniku prikaz statistike za 2023.godinu. Povezan je s funkcijom „prikazi_statistiku_2023“.

Slično tome, treći gumb naziva „Prikaži statistiku za 2024“ omogućuje korisniku prikaz statistike za 2024. godinu. Povezan je s funkcijom „prikazi_statistiku_2024“.

```
298 class StatistikaApp(QMainWindow):
299     def __init__(self, parent=None):
300         super().__init__(parent)
301         self.setWindowTitle("Statistika 2023")
302         self.setGeometry(100, 100, 400, 400)
303         main_widget = QWidget()
304         main_layout = QVBoxLayout()
305         main_widget.setLayout(main_layout)
306         self.setCentralWidget(main_widget)
307
308         button_styles = '''
309             QPushButton {
310                 background-color: #ff6347; /* Promijenite boju ovdje */
311                 color: white;
312                 border: none;
313                 border-radius: 5px;
314                 padding: 10px;
315                 font-size: 16px;
316             }
317             QPushButton:hover {
318                 background-color: #ff4d26; /* Promijenite boju ovdje */
319             }
320         '''
321
322         self.r2_mse_button = QPushButton("Prikaži R2 i MSE", self)
323         self.r2_mse_button.setStyleSheet(button_styles)
324         self.r2_mse_button.clicked.connect(self.r2_mse)
325
326         self.statistika_2023_button = QPushButton("Prikaži statistiku za 2023", self)
327         self.statistika_2023_button.setStyleSheet(button_styles)
328         self.statistika_2023_button.clicked.connect(self.prikazi_statistiku_2023)
329
330         self.statistika_2024_button = QPushButton("Prikaži statistiku za 2024", self)
331         self.statistika_2024_button.setStyleSheet(button_styles)
332         self.statistika_2024_button.clicked.connect(self.prikazi_statistiku_2024)
333
334         self.result_label = QLabel(self)
335
336         self.result_text = QTextEdit(self)
337         self.result_text.setReadOnly(True)
338
339         main_layout.addWidget(self.r2_mse_button)
340         main_layout.addWidget(self.statistika_2023_button)
341         main_layout.addWidget(self.statistika_2024_button)
342         main_layout.addWidget(self.result_label)
343         main_layout.addWidget(self.result_text)
```

Slika 23. Prikaz koda sučelja Statistika

Izvor: Autor



Slika 24. Prikaz sučelja Statistika

Izvor: Autor

6.2. Funkcije statistika sučelja

Kad je korisnik kliknuo na gumb „Prikaži R² i MSE“, poziva se funkcija `r2_mse` i na korisničkom sučelju prikazuje mu se vrijednosti koeficijenta determinacije (r^2) i srednje kvadratne greške (`mse`).

```
297 def r2_mse(self):
298     self.result_label.setText(f"koeficijent determinacije i srednja kvadratna greška: {r2}, {mse}")
299     pass
```

Slika 25. Prikaz koda funkcije `r2_mse`

Izvor: Autor

```
Koeficijent determinacije i Srednja kvadratna greška: 0.9772918039864524, 1.550027301975588
```

Slika 26. Prikaz R² i MSE

Izvor: Autor

Kad je korisnik kliknuo gumb „Prikaži statistiku za 2023“, poziva se funkcija `prikazi_statistiku_2023`. Ova funkcija pruža korisniku uvid u prosječne vrijednosti parametara i detalje o trenutnom vremenu, rezultati se prikazuju na korisničkom sučelju.

Funkcija prolazi kroz svaki parametar. Ako je trenutni parametar 'Trenutno vrijeme', tada se za svako mjesto izračunavaju postoci za različite vremenske uvjete ,kao što su kiša, oblačno, itd. A ako trenutni parametar nije 'Trenutno vrijeme', tada se za svako pojedinačno mjesto računa prosječna vrijednost tog parametra.

Ovo korisnicima omogućuje bolje razumijevanje klimatskih uvjeta u 2023. godini.

```
301 def prikazi_statistiku_2023(self):
302     parameters = [
303         'TrenutačnaTemperatura', 'Oborine', 'Vlaga', 'Vjetar', 'Trenutnovrijeme',
304         'Najvećatemperatura', 'Najmanjatemperatura'
305     ]
306     places = df['Mjesto'].unique()
307     self.result_text.clear()
308     for parameter in parameters:
309         if parameter == 'Trenutnovrijeme':
310             for place in places:
311                 df_place = df.loc[df['Mjesto'] == place]
312                 postotak_kisa = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 0 else 0).mean() * 100
313                 postotak_oblacno = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 1 else 0).mean() * 100
314                 postotak_poluoblacno = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 2 else 0).mean() * 100
315                 postotak_slabipljuskovi = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 3 else 0).mean() * 100
316                 postotak_suncano = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 4 else 0).mean() * 100
317                 postotak_uglavnomoblacno = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 5 else 0).mean() * 100
318                 postotak_uglavnom_suncano = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 6 else 0).mean() * 100
319                 postotak_vedro = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 7 else 0).mean() * 100
320                 postotak_vedronaoblaka = df_place['Trenutnovrijeme'].apply(lambda x: 1 if x == 8 else 0).mean() * 100
321                 self.result_text.append(f'Postotak kišnih dana u {place}: {postotak_kisa:.2f}%\n')
322                 self.result_text.append(f'Postotak oblačnih dana u {place}: {postotak_oblacno:.2f}%\n')
323                 self.result_text.append(f'Postotak poluoblačnih dana u {place}: {postotak_poluoblacno:.2f}%\n')
324                 self.result_text.append(f'Postotak slabih pljuskova dana u {place}: {postotak_slabipljuskovi:.2f}%\n')
325                 self.result_text.append(f'Postotak sunčanih dana u {place}: {postotak_suncano:.2f}%\n')
326                 self.result_text.append(f'Postotak uglavnom oblačnih dana u {place}: {postotak_uglavnomoblacno:.2f}%\n')
327                 self.result_text.append(f'Postotak uglavnom sunčanih dana u {place}: {postotak_uglavnom_suncano:.2f}%\n')
328                 self.result_text.append(f'Postotak vedrih dana u {place}: {postotak_vedro:.2f}%\n')
329                 self.result_text.append(f'Postotak vedro s naoblakom dana u {place}: {postotak_vedronaoblaka:.2f}%\n')
330                 self.result_text.append('\n')
331         else:
332             for place in places:
333                 df_place = df.loc[df['Mjesto'] == place]
334                 mean_value = df_place[parameter].mean()
335                 self.result_text.append(f'Prosječna vrijednost parametra {parameter} u {place}: {mean_value:.2f}%\n')
```

Slika 27. Prikaz koda funkcije `prikazi_statistiku_2023`

Izvor: Autor

```

Prosječna vrijednost parametra TrenutačnaTemperatura u Čakovec: 11.53
Prosječna vrijednost parametra TrenutačnaTemperatura u Varaždin: 11.70
Prosječna vrijednost parametra TrenutačnaTemperatura u Koprivnica: 11.97

Prosječna vrijednost parametra Oborine u Čakovec: 36.40
Prosječna vrijednost parametra Oborine u Varaždin: 34.30
Prosječna vrijednost parametra Oborine u Koprivnica: 28.47

Prosječna vrijednost parametra Vlaga u Čakovec: 67.93
Prosječna vrijednost parametra Vlaga u Varaždin: 67.47
Prosječna vrijednost parametra Vlaga u Koprivnica: 68.00

Prosječna vrijednost parametra Vjetar u Čakovec: 9.13
Prosječna vrijednost parametra Vjetar u Varaždin: 8.97
Prosječna vrijednost parametra Vjetar u Koprivnica: 9.63

Postotak kišnih dana u Čakovec: 3.33%
Postotak oblačnih dana u Čakovec: 10.00%
Postotak poluoblačnih dana u Čakovec: 33.33%
Postotak slabih pljuskova dana u Čakovec: 20.00%
Postotak sunčanih dana u Čakovec: 6.67%
Postotak uglavnom oblačnih dana u Čakovec: 13.33%
Postotak uglavnom sunčanih dana u Čakovec: 6.67%
Postotak vedrih dana u Čakovec: 3.33%
Postotak vedro s naoblakom dana u Čakovec: 3.33%

```

Slika 28. Prikaz statistike za 2023.godinu.

Izvor: Autor

Kad je korisnik kliknuo gumb „Prikaži statistiku za 2024“, poziva se funkcija `prikazi_statistiku_2024`.

Ova funkcija ima sličnu logiku kao funkcija „`prikazi_statistiku_2023`“, ali prikazuje statistiku za 2024. godinu na temelju drugog DataFrame-a ("`new_df_2024`"). Rezultati se prikazuju na korisničkom sučelju.

349

```
df_place = new_df_2024.loc[df['Mjesto'] == place]
```

Slika 29. Prikaz dijela koda s promijenjenim DataFrame za 2024.godinu.

Izvor: Autor

Prosječna vrijednost parametra TrenutačnaTemperatura u Čakovec: 11.60
Prosječna vrijednost parametra TrenutačnaTemperatura u Varaždin: 11.60
Prosječna vrijednost parametra TrenutačnaTemperatura u Koprivnica: 12.00
Prosječna vrijednost parametra Oborine u Čakovec: 36.47
Prosječna vrijednost parametra Oborine u Varaždin: 34.00
Prosječna vrijednost parametra Oborine u Koprivnica: 28.50
Prosječna vrijednost parametra Vlaga u Čakovec: 68.03
Prosječna vrijednost parametra Vlaga u Varaždin: 67.23
Prosječna vrijednost parametra Vlaga u Koprivnica: 68.10
Prosječna vrijednost parametra Vjetar u Čakovec: 9.13
Prosječna vrijednost parametra Vjetar u Varaždin: 8.97
Prosječna vrijednost parametra Vjetar u Koprivnica: 9.63
Postotak kišnih dana u Čakovec: 3.33%
Postotak oblačnih dana u Čakovec: 13.33%
Postotak poluoblačnih dana u Čakovec: 30.00%
Postotak slabih pljuskova dana u Čakovec: 20.00%
Postotak sunčanih dana u Čakovec: 6.67%
Postotak uglavnom oblačnih dana u Čakovec: 13.33%
Postotak uglavnom sunčanih dana u Čakovec: 6.67%
Postotak vedrih dana u Čakovec: 3.33%
Postotak vedro s naoblakom dana u Čakovec: 3.33%

Slika 30. Prikaz statistike za 2024. godinu.

Izvor: Autor

6.3. Vrijeme sučelje

Kad je korisnik u glavnom sučelju kliknuo na gumb „Vrijeme“, otvorit će mu se sučelje s prikazom tablice i vremena za svako pojedino mjesto, za mjesec travanj 2023. i 2024. godine.

Korisnik ne može vidjeti vrijeme ako nije odabrao sve excel datoteke i napravio treniranje i testiranje, prikazat će mu se greška s porukom.

Sučelje prikazuje naziv trenutnog mjesta, tablicu s vremenskim podacima, i gumb za prijelaz na sljedeće mjesto s dodanim stilom.

```
428 class PrvoSučelje(QMainWindow):
429     def __init__(self,df,new_df_2024,places,parent=None):
430         super().__init__(parent)
431
432         self.places = places
433         self.df = df
434         self.new_df_2024 = new_df_2024
435         self.places = places
436         self.current_place_index = 0
437         self.setWindowTitle("Vrijeme")
438         self.central_widget = QWidget()
439         self.setCentralWidget(self.central_widget)
440         self.layout = QVBoxLayout()
441         self.central_widget.setLayout(self.layout)
442
443         button_styles = '''
444             QPushButton {
445                 background-color: #ff6347; /* Promijenite boju ovdje */
446                 color: white;
447                 border: none;
448                 border-radius: 5px;
449                 padding: 10px;
450                 font-size: 16px;
451             }
452             QPushButton:hover {
453                 background-color: #ff4d26; /* Promijenite boju ovdje */
454             }
455             ...
456
457         self.place_label = QLabel()
458         self.layout.addWidget(self.place_label)
459
460         self.table = QTableWidgetItem()
461         self.layout.addWidget(self.table)
462
463         self.next_button = QPushButton("Sljedeće mjesto")
464         self.next_button.setStyleSheet(button_styles)
465         self.next_button.clicked.connect(self.show_next_place)
466         self.layout.addWidget(self.next_button)
467         self.show_weather()
```

Slika 31. Prikaz koda sučelja Vrijeme.

Izvor: Autor

6.3.1. Funkcije vrijeme sučelja

Funkcija `show_weather` se koristi za prikazivanje vremenskih podataka u tablici, definiraju se početni i završni datumi za 2023. i 2024. godinu te se iz DataFrame-ova: `df` i `new_df_2024` filtriraju podaci za odabrano mjesto u tim godinama, dodani su i smajlići koji odgovaraju za svako vrijeme.

Vremenski podaci se prikazuju u tablici, gdje se svaki redak sastoji od datuma i odgovarajućih vremenskih podataka za 2023. i 2024. godinu.

Ako se vremenski podaci za te dvije godine razlikuju, tekst se poveća i podeblja kako bi se primijetila razlika.

```

409 def show_weather(self):
410     place = self.places[self.current_place_index]
411     start_date_2023 = datetime(2023, 4, 1)
412     end_date_2023 = datetime(2023, 4, 30)
413     new_df_place_2023 = self.df.loc[self.df['Mjesto'] == place].assign(Datum=pd.date_range(start=start_date_2023, end=end_date_2023))
414
415     start_date_2024 = datetime(2024, 4, 1)
416     end_date_2024 = datetime(2024, 4, 30)
417     new_df_place_2024 = self.new_df_2024.loc[self.new_df_2024['Mjesto'] == place].assign(Datum=pd.date_range(start=start_date_2024, end=end_date_2024))
418
419     trenutnovrijeme_dict = {
420         0: 'Kiša', 1: 'Oblačno', 2: 'Poluoblačno', 3: 'Slabi pljuskovi uz kišu',
421         4: 'Sunčano', 5: 'Uglavnom oblačno', 6: 'Uglavnom sunčano',
422         7: 'Vedro', 8: 'Vedro s povremenom naoblakom'
423     }
424
425     smajlici = {
426         'Kiša': '☔', 'Oblačno': '☁', 'Poluoblačno': '⛅',
427         'Slabi pljuskovi uz kišu': '🌧', 'Sunčano': '☀', 'Uglavnom oblačno': '☁',
428         'Uglavnom sunčano': '☀', 'Vedro': '☀', 'Vedro s povremenom naoblakom': '☀'
429     }
430
431     self.place_label.setText(place)
432
433     header_labels = ['Datum', '2023', f'2024 - {place}']
434     self.table.setColumnCount(len(header_labels))
435     self.table.setHorizontalHeaderLabels(header_labels)
436
437     self.table.setRowCount(len(new_df_place_2023))
438     self.table.verticalHeader().setVisible(False)
439
440     for row, (_, row_2023), (_, row_2024) in zip(range(len(new_df_place_2023)), new_df_place_2023.iterrows(), new_df_place_2024.iterrows()):
441         godina_str = row_2023['Datum'].strftime('%d.%m')
442         godina_str_2023 = row_2023['Datum'].strftime('%Y')
443         godina_str_2024 = row_2024['Datum'].strftime('%Y')
444
445         trenutnovrijeme_2023 = trenutnovrijeme_dict.get(row_2023['Trenutnovrijeme'], '')

```

Slika 32. Prikaz dijela koda funkcije show_weather.

Izvor: Autor

Funkcija show_next_place se koristi za prijelaz na sljedeće mjesto s pritiskom na gumb ažurira se trenutni indeks mjesta na sljedeći indeks u listi places i ponovno se poziva show_weather metoda za prikaz novih vremenskih podataka.

```

469 def show_next_place(self):
470     self.current_place_index = (self.current_place_index + 1) % len(self.places)
471     self.show_weather()

```

Slika 33. Prikaz koda funkcije show_next_place.

Izvor: Autor

Vrijeme

Čakovec

Datum	2023	2024 - Čakovec
01.04	 Poluoblačno	 Poluoblačno
02.04	 Poluoblačno	 Poluoblačno
03.04	 Oblačno	 Oblačno
04.04	 Kiša	 Kiša
05.04	 Vedro s povremenom naoblakom	 Vedro s povremenom naoblakom
06.04	 Uglavnom sunčano	 Uglavnom sunčano
07.04	 Uglavnom oblačno	 Uglavnom oblačno
08.04	 Poluoblačno	 Oblačno
09.04	 Slabi pljuskovi uz kišu	 Slabi pljuskovi uz kišu
10.04	 Slabi pljuskovi uz kišu	 Slabi pljuskovi uz kišu
11.04	 Slabi pljuskovi uz kišu	 Slabi pljuskovi uz kišu
12.04	 Slabi pljuskovi uz kišu	 Slabi pljuskovi uz kišu
13.04	 Uglavnom oblačno	 Uglavnom oblačno
14.04	 Slabi pljuskovi uz kišu	 Slabi pljuskovi uz kišu
15.04	 Oblačno	 Oblačno
16.04	 Poluoblačno	 Poluoblačno
17.04	 Poluoblačno	 Poluoblačno
18.04	 Vedro	 Vedro
19.04	 Uglavnom oblačno	 Uglavnom oblačno
20.04	 Oblačno	 Oblačno
21.04	 Sunčano	 Sunčano
22.04	 Poluoblačno	 Poluoblačno
23.04	 Poluoblačno	 Poluoblačno
24.04	 Uglavnom oblačno	 Uglavnom oblačno
25.04	 Uglavnom sunčano	 Uglavnom sunčano
26.04	 Poluoblačno	 Poluoblačno
27.04	 Poluoblačno	 Poluoblačno
28.04	 Slabi pljuskovi uz kišu	 Slabi pljuskovi uz kišu
29.04	 Poluoblačno	 Poluoblačno
30.04	 Sunčano	 Sunčano

Slika 34. Prikaz sučelja vrijeme za mjesto Čakovec.

Izvor: Autor

6.3. Vremenska prognoza sučelje

Kad je korisnik na glavnom sučelju kliknuo na gumb „Vremenska prognoza“, otvara mu se sučelje s vremenskom prognozom. Ako korisnik nije odabrao sve excel datoteke i napravio treniranje i testiranje, neće moći dobiti prikaz vremenske prognoze, već će mu se prikazati poruka s greškom.

Sučelje koristi tkinter. Autor je odabrao tkinter zbog ljepšeg i vizualnijeg prikaza vremenske prognoze na korisničkom sučelju s različitim grafovima. Stvara glavni okvir koji sadržava grafove, scrollbar kako bi se mogli prikazati svi grafovi i klizati kroz sadržaj.

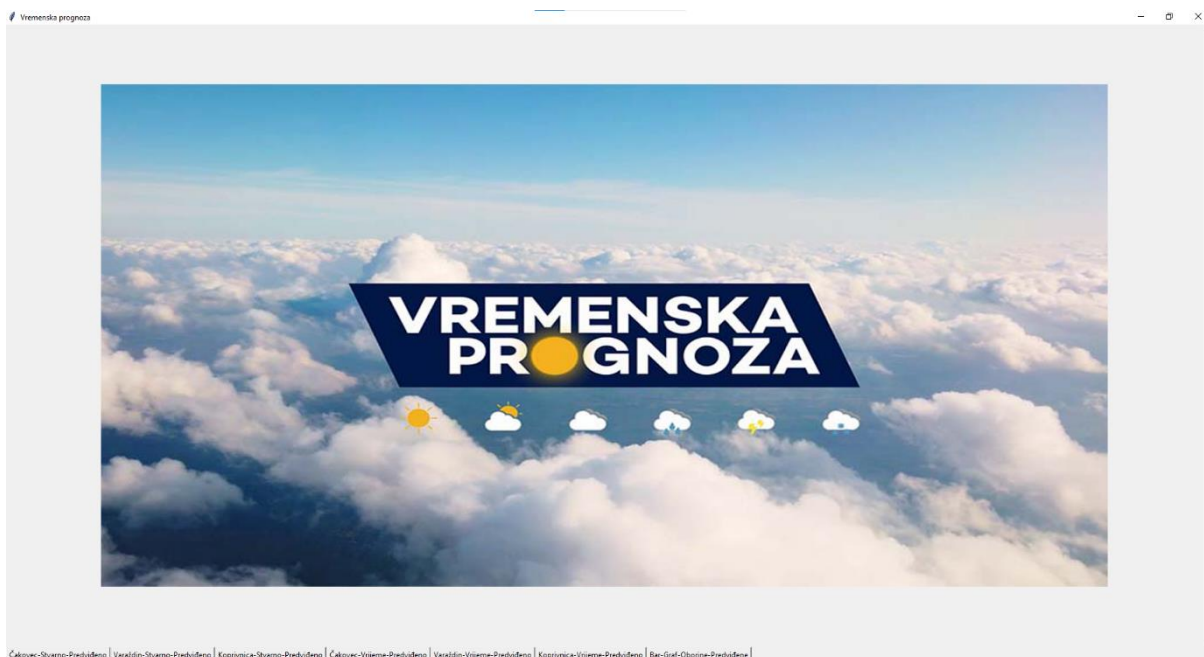
Canvas je stvoren unutar glavnog okvira (main_frame) i popunjava preostali prostor. Unutar canvasa, stvara se unutarnji okvir (inner_frame) koji sadržava grafove, dodana je i slika kako bi sučelje izgledalo ljepše.

Na sučelju se nalazi 7 gumbova, povezana s odgovarajućom funkcijom koja će se izvršiti kada se gumb klikne. To su „Čakovec-Stvarno-Predviđeno“, „Varaždin-Stvarno-Predviđeno“, „Koprivnica-Stvarno-Predviđeno“, „Čakovec-Vrijeme-Predviđeno“, „Varaždin-Vrijeme-Predviđeno“, „Koprivnica-Vrijeme-Predviđeno“, „Bar-Graf-Oborine-Predviđene“.

```
538 def third_interface():
539     root = tk.Tk()
540     root.title("Vremenska prognoza")
541
542
543     main_frame = tk.Frame(root)
544     main_frame.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
545
546
547     scrollbar = ttk.Scrollbar(main_frame, orient=tk.VERTICAL)
548     scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
549
550
551     canvas = tk.Canvas(main_frame, yscrollcommand=scrollbar.set)
552     canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)
553
554     canvas.configure(yscrollcommand=scrollbar.set)
555
556
557     inner_frame = tk.Frame(canvas)
558     canvas.create_window((0, 0), window=inner_frame, anchor='nw')
559
560     scrollbar.configure(command=canvas.yview)
561     canvas.configure(yscrollcommand=scrollbar.set)
562
563
564     image = Image.open('C:/Users/fbara/Desktop/vremenska-prognoza-1.jpg')
565     image = image.resize((1600, 800))
566     photo = ImageTk.PhotoImage(image)
567     label = tk.Label(canvas, image=photo)
568     label.pack(fill="both", expand=True)
```

Slika 35. Prikaz koda Vremenska prognoza sučelja.

Izvor: Autor



Slika 36. Prikaz Vremenska prognoza sučelja.

Izvor: Autor

6.3.1. Vremenska prognoza sučelje funkcije

Kad je korisnik kliknuo gumb „Čakovec-Stvarno-Predviđeno" poziva se funkcija `otvori_graf('Čakovec')`.

Grafovi se generiraju funkcijom `stvar_pred_graph()` koja prikazuje sve grafove za mjesto Čakovec. Prikazuje ukupno 7 grafova, za svaki parametar se prikazuje jedan graf. Grafovi se stvaraju unutar okvira (`inner_frame`), a podaci se dobivaju iz `DataFrame`-ova: `df` i `new_df`. Unutar funkcije, definiran je rječnik `units` koji sadrži jedinice za parametre kao ključeve i koristi se za prikaz jedinica na osi y grafa. Dalje se koristi `FigureCanvasTkAgg` kako bi se graf prikazao unutar tkinter sučelja.

Slično tome, kad je korisnik kliknuo na gumbove „Varaždin-Stvarno-Predviđeno" i „Koprivnica-Stvarno-Predviđeno", poziva istu funkciju `otvori_graf()`. Jedina je razlika da se grafovi prikazuju za mjesto Varaždin i Koprivnica, dok sve ostalo ostaje isto.


```
518     def stvar_pred_graph(inner_frame, df, new_df, place, parameter, ylabel):
519         units = {
520             'Oborine': '%',
521             'Vlaga': '%',
522             'Vjetar': 'km/h',
523             'Najvećatempertura': '°C',
524             'Najmanjatempertura': '°C'
525         }
526     }
527     fig = plt.figure(figsize=(10, 5))
528     df_place = df.loc[df['Mjesto'] == place]
529     new_df_place = new_df.loc[new_df['Mjesto'] == place]
530     plt.plot(new_df_place['Datum'], df_place[parameter], label='Stvarne vrijednosti')
531     plt.plot(new_df_place['Datum'], new_df_place[parameter], label='Predvidene vrijednosti')
532     plt.xlabel('Datum')
533     plt.ylabel(f'{parameter} ({units[parameter]})')
534     plt.title('Stvarne i predvidene vrijednosti za {} - {}'.format(place, parameter))
535     plt.legend()
536     plt.close()
537     canvas = FigureCanvasTkAgg(fig, master=inner_frame)
538     canvas.draw()
539     canvas.get_tk_widget().pack()
```

Slika 37. Prikaz koda funkcije stvar_pred_graph().

Izvor: Autor

„Funkcija prikazi_grafove(place)“ generira grafove za odabrano mjesto. Nakon poziva ove funkcije, prethodno prikazani grafovi će biti uklonjeni, a zatim će se za svaki parametar iz liste parameters generirati graf unutar inner_frame. Prvo se koristi label.pack_forget() kako bi se uklonio prethodno prikazani tekstualni sadržaj.

Ako je odabrano mjesto Čakovec, definiraju se početni i krajnji datum za 2024. godinu te se filtrira DataFrame: new_df, samo za mjesto Čakovec i dodaje se stupac 'Datum' koji predstavlja raspon datuma za taj period. Zatim se poziva funkcija stvar_pred_graph() s odgovarajućim argumentima kako bi se generirao graf za to mjesto, parametar i podatke.

A ako nije odabrano ni 'Varaždin' ni 'Koprivnica', podaci za 2024. godinu se uzimaju iz DataFrame-a: new_df_2024. Zatim se poziva funkcija stvar_pred_graph() kako bi se generirao graf za to mjesto, parametar i podatke.

```
543 parameters = ['Oborine', 'Vlaga', 'Vjetar', 'Najvećatemperatura', 'Najmanjatemperatura']
544 places = ['Čakovec', 'Varaždin', 'Koprivnica']
545
546 def prikazi_grafove(place):
547     label.pack_forget()
548     for parameter in parameters:
549         inner_frame_graph = tk.Frame(inner_frame)
550         inner_frame_graph.pack()
551
552         if place == 'Varaždin':
553             start_date = datetime(2024, 4, 1)
554             end_date = datetime(2024, 4, 30)
555             new_df_varazdin = new_df.loc[new_df['Mjesto'] == 'Varaždin']
556             new_df_varazdin = new_df_varazdin.assign(Datum=pd.date_range(start=start_date, end=end_date, freq='D'))
557             stvar_pred_graph(inner_frame_graph, df, new_df_varazdin, place, parameter, ylabel=parameter)
558         elif place == 'Koprivnica':
559             start_date = datetime(2024, 4, 1)
560             end_date = datetime(2024, 4, 30)
561             new_df_koprivnica = new_df.loc[new_df['Mjesto'] == 'Koprivnica']
562             new_df_koprivnica = new_df_koprivnica.assign(Datum=pd.date_range(start=start_date, end=end_date, freq='D'))
563             stvar_pred_graph(inner_frame_graph, df, new_df_koprivnica, place, parameter, ylabel=parameter)
564         else:
565             stvar_pred_graph(inner_frame_graph, df, new_df_2024, place, parameter, ylabel=parameter)
566
```

Slika 38. Prikaz koda funkcije prikazi_grafove().

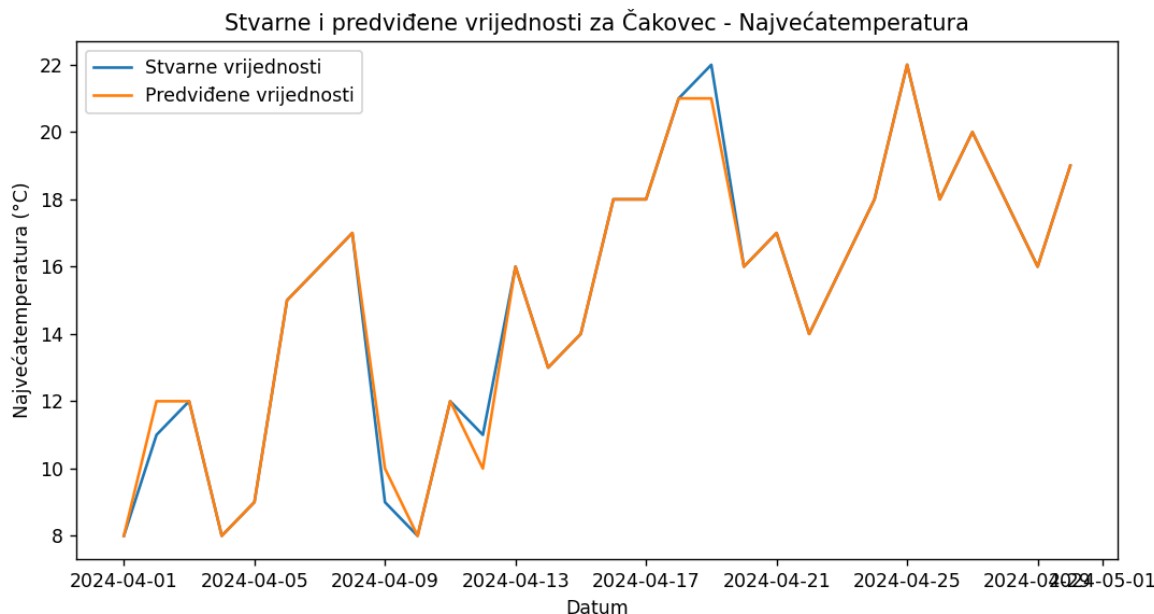
Izvor: Autor

„Funkcija otvori_graf(mjesto)“ otvara grafove za odabrano mjesto. Nakon poziva ove funkcije, prethodno prikazani grafovi će biti uklonjeni, a zatim će se pozvati funkcija prikazi_grafove(mjesto).

```
569     def otvori_graf(mjesto):
570         label.pack_forget()
571
572         for widget in inner_frame.winfo_children():
573             widget.destroy()
574
575         prikazi_grafove(mjesto)|
576
```

Slika 39. Prikaz koda funkcije otvori_graf(mjesto).

Izvor: Autor



Slika 40. Prikaz grafa s parametrom najveća temperatura za mjesto Čakovec.

Izvor: Autor

Kad je korisnik kliknuo gumb „Varaždin-Vrijeme-Predviđeno" poziva se funkcija `otvori_graff(„Varaždin“)`.

Graf se generira funkcijom `prikazi_trenutno_vrijeme()` i prikazuje jedan graf, usporedbu vremena 2023. i 2024. godine za mjesec travanj. Unutar funkcije definiran je rječnik trenutno vrijeme_dict koji zamjenjuje numeričke vrijednosti trenutnog vremena na odgovarajući tekstualni opis. Zatim se iz DataFrame-a: `df` i `new_df_2024` filtriraju podaci za odabrano mjesto i zadržavaju se samo stupci „Datum“ i „Trenutno vrijeme“. Nakon toga, podaci se spajaju na temelju stupca „Datum“ koristeći `merge()` funkciju.

Slično tome, kad je korisnik kliknuo na gumbove „Čakovec-Vrijeme-Predviđeno" i „Koprivnica-Vrijeme-Predviđeno" poziva istu funkciju `otvori_graff()`. Razlika je u tome da se grafovi prikazuju za mjesto Čakovec i Koprivnica.

```

582 def prikazi_trenutno_vrijeme(df, new_df_2024, place):
583     trenutnovrijeme_dict = {
584         0: 'Kiša', 1: 'Oblačno', 2: 'Poluoblačno', 3: 'Slabi pljuskovi uz kišu',
585         4: 'Sunčano', 5: 'Uglavnom oblačno', 6: 'Uglavnom sunčano',
586         7: 'Vedro', 8: 'Vedro s povremenom naoblakom'
587     }
588
589     trenutno_vrijeme_2023 = df.loc[df['Mjesto'] == place, ['Datum', 'Trenutnovrijeme']]
590     trenutno_vrijeme_2024 = new_df_2024.loc[new_df_2024['Mjesto'] == place, ['Datum', 'Trenutnovrijeme']]
591
592     merged_df = trenutno_vrijeme_2023.merge(trenutno_vrijeme_2024, on='Datum', how='inner')
593
594     fig = plt.figure(figsize=(12, 6))
595
596     plt.plot(merged_df['Datum'], [trenutnovrijeme_dict.get(v, '') for v in merged_df['Trenutnovrijeme_x']], 'bo-', label=
597     plt.plot(merged_df['Datum'], merged_df['Trenutnovrijeme_y'], 'ro-', label='2024')
598
599     plt.xlabel('Datum')
600     plt.ylabel('Trenutno vrijeme')
601     plt.title(f"Trenutno vrijeme u {place} - Usporedba 2023. i 2024.")
602     plt.legend()
603
604     plt.xticks(rotation=45)
605
606     plt.tight_layout()
607     plt.close()
608     canvas = FigureCanvasTkAgg(fig, master=inner_frame)
609     canvas.draw()
610     canvas.get_tk_widget().pack(fill=tk.BOTH, expand=True)

```

Slika 41. Prikaz koda funkcije prikazi_trenutno_vrijeme().

Izvor: Autor

Funkcija `otvori_graff(mjesto)` služi za otvaranje grafa za odabrano mjesto. Nakon poziva ove funkcije, prethodno prikazani grafovi će biti uklonjeni, a zatim će se za odabrano mjesto generirati graf koji prikazuje usporedbu trenutnog vremena između 2023. i 2024. godine.

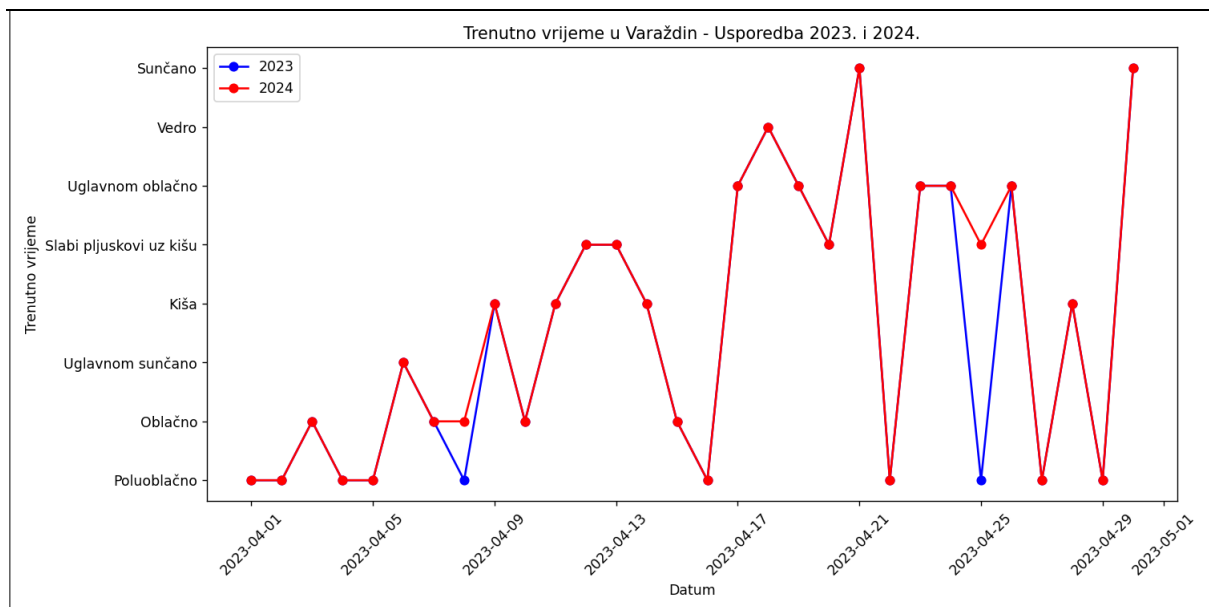
```

678 def otvori_graff(mjesto):
679     label.pack_forget()
680     for widget in inner_frame.winfo_children():
681         widget.destroy()
682
683     start_date = datetime(2023, 4, 1)
684     end_date = datetime(2023, 4, 30)
685     new_df_place_2023 = df.loc[df['Mjesto'] == mjesto].assign(Datum=pd.date_range(start=start_date, end=end_date, freq=
686
687     start_date_2024 = datetime(2024, 4, 1)
688     end_date_2024 = datetime(2024, 4, 30)
689     new_df_place_2024 = new_df_2024.loc[new_df_2024['Mjesto'] == mjesto].assign(Datum=pd.date_range(start=start_date, e
690
691     prikazi_trenutno_vrijeme(new_df_place_2023, new_df_place_2024, mjesto)

```

Slika 42. Prikaz koda funkcije otvori_graff().

Izvor: Autor



Slika 43. Prikaz grafa usporedba 2023 i 2024 za mjesto Varaždin.

Izvor: Autor

Kad je korisnik kliknuo gumb „Bar-Graf-Oborine-Predviđene” poziva se funkcija prikazi_treci_graf.

Funkcija prikazi_totalne_oborine_graf() generira graf za mjesta Čakovec, Varaždin, Koprivnica i prikazuje ukupnu količinu oborina koja je pala u mjesecu travnju 2023. te kolika je ukupna predviđena količina oborina za mjesec travanj 2024.godine. Grafovi se generiraju pomoću plt.bar() funkcije. Prva traka predstavlja količine oborina za 2023. godinu, a druga traka predstavlja količine oborina za 2024. godinu. Također se dodaju tekstualne oznake iznad svake trake s količinom oborina pomoću plt.text() funkcije.

```

632 def prikazi_totalne_oborine_graf(inner_frame, total_rain_2023, total_rain_2024):
633     label.pack_forget()
634     bar_width = 0.35
635     r1 = np.arange(len(total_rain_2023))
636     r2 = [x + bar_width for x in r1]
637
638     fig = plt.figure(figsize=(10, 5))
639
640     plt.bar(r1, total_rain_2023.values, color='b', width=bar_width, edgecolor='white', label='2023')
641     plt.bar(r2, total_rain_2024.values, color='r', width=bar_width, edgecolor='white', label='2024')
642     plt.xlabel('Grad')
643     plt.ylabel(f'Oborine u mm')
644     plt.title('Ukupne količine oborina po gradu')
645     plt.xticks([r + bar_width / 2 for r in range(len(total_rain_2023))], total_rain_2023.index)
646     for i, v in enumerate(total_rain_2023.values):
647         plt.text(r1[i] - 0.08, v + 1, str(v), color='blue', fontweight='bold')
648
649     for i, v in enumerate(total_rain_2024.values):
650         plt.text(r2[i] - 0.08, v + 1, str(v), color='red', fontweight='bold')
651
652     plt.close()
653     canvas = FigureCanvasTkAgg(fig, master=inner_frame)
654     canvas.draw()
655     canvas.get_tk_widget().pack()
656

```

Slika 44. Prikaz koda funkcije prikazi_totalne_oborine.

Izvor: Autor

Funkcija `prikazi_treci_graf()` služi za prikazivanje trećeg grafa koji prikazuje ukupne količine oborina po mjestu za mjesec travanj 2023. i 2024. godine. Iz izvornog DataFramea `df` i DataFramea za 2024. godinu `new_df_2024`, računaju se ukupne količine oborina po mjestima.

Grupiranje i zbrajanje oborina vrši se prema stupcu „Mjesto“. Rezultati se pohranjuju u `total_rain_2023` i `total_rain_2024`. Na kraju se poziva funkcija `prikazi_totalne_oborine_graf()` kako bi se prikazao graf ukupnih količina oborina po mjestu za mjesec travanj 2023. i 2024.godine.

```
657     def prikazi_treci_graf():
658         label.pack_forget()
659         for widget in inner_frame.winfo_children():
660             widget.destroy()
661         total_rain_2023 = df.groupby('Mjesto')['Oborine'].sum()
662         total_rain_2024 = new_df_2024.groupby('Mjesto')['Oborine'].sum()
663         prikazi_totalne_oborine_graf(inner_frame, total_rain_2023, total_rain_2024)
664
```

Slika 45. Prikaz koda funkcije `prikazi_treci_graf()`.

Izvor: Autor

```
668     graf_cakovec_button = tk.Button(root, text="Čakovec-Stvarno-Predviđeno", command=lambda: otvori_graf('Čakovec'))
669     graf_cakovec_button.pack(side=tk.LEFT)
670
671     graf_varazdin_button = tk.Button(root, text="Varaždin-Stvarno-Predviđeno", command=lambda: otvori_graf('Varaždin'))
672     graf_varazdin_button.pack(side=tk.LEFT)
673
674     graf_koprivnica_button = tk.Button(root, text="Koprivnica-Stvarno-Predviđeno", command=lambda: otvori_graf('Koprivnica'))
675     graf_koprivnica_button.pack(side=tk.LEFT)
676     cakovec_button = tk.Button(root, text="Čakovec-Vrijeme-Predviđeno", command=lambda: otvori_graff('Čakovec'))
677     cakovec_button.pack(side=tk.LEFT)
678
679     varazdin_button = tk.Button(root, text="Varaždin-Vrijeme-Predviđeno", command=lambda: otvori_graff('Varaždin'))
680     varazdin_button.pack(side=tk.LEFT)
681
682     koprivnica_button = tk.Button(root, text="Koprivnica-Vrijeme-Predviđeno", command=lambda: otvori_graff('Koprivnica'))
683     koprivnica_button.pack(side=tk.LEFT)
684     bar_button = tk.Button(root, text="Bar-Graf-Oborine-Predviđene", command=prikazi_treci_graf)
685     bar_button.pack(side=tk.LEFT)
686
687
688     canvas.configure(scrollregion=canvas.bbox("all"), yscrollcommand=scrollbar.set)
689
690
691     inner_frame.bind("<Configure>", lambda e: canvas.configure(scrollregion=canvas.bbox("all")))
692
693
694     scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
695
696     root.mainloop()
```

Slika 46. Prikaz koda za stvaranje i prikazivanje gumba na sučelju.

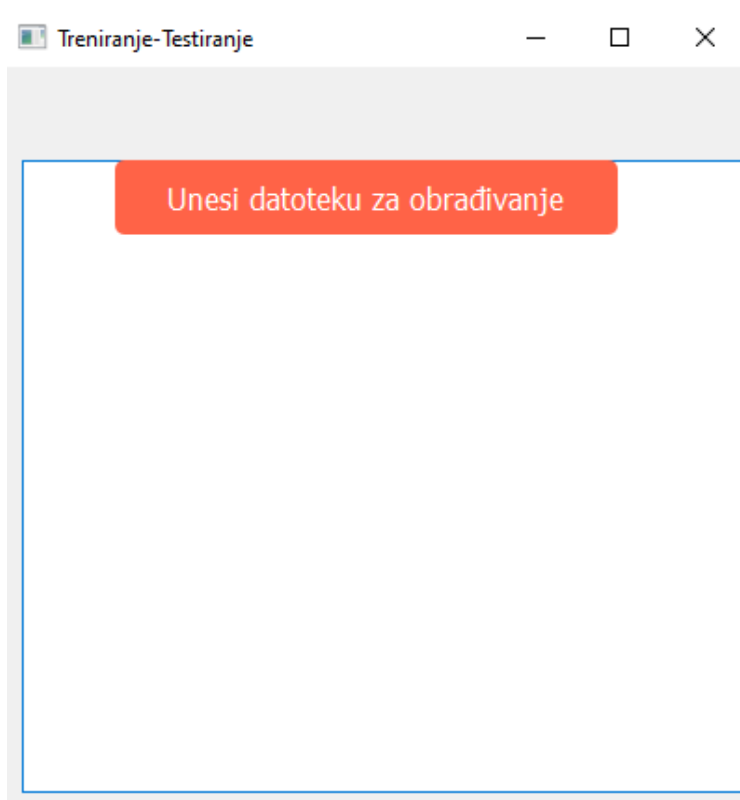
Izvor: Autor

Kod predstavlja stvaranje i prikazivanje gumba na korisničkom sučelju i konfiguriranje prikaza trake za pomicanje (scrollanje) na korisničkom sučelju. Funkcija `root.mainloop()` pokreće glavnu petlju aplikacije kako bi se prikazali gumbi i korisničko sučelje i omogućilo interakciju s njima.

6.4. Treniranje-testiranje-pojedinačno

Kad je korisnik u glavnom sučelju kliknuo na gumb „Treniranje i testiranje-pojedinačno“, otvorit će mu se sučelje u kojem se koristi glavni prozor, tekstni editor i gumb pod nazivom „Unesi datoteku za treniranje i testiranje“ s dodanim stilom.

Ovo sučelje je slično sučelju Treniranje i testiranje i ima sličnu logiku i funkciju koju je autor već naveo i opisao. Razlika je u tome što korisnik može odabrati excel podatke za bilo koje mjesto, bilo koji mjesec za 2023.godinu, s istim parametrima. Izvršit će mu se treniranje i testiranje i predviđeni podaci za 2024. godinu prikazuju se u tekstualnom editoru.



Slika 47. Prikaz sučelja treniranje-testiranje-pojedinačno

Izvor: Autor

```
808     def process_files(self, filename):
809         self.button.hide()
810         global df, new_df
811         df = pd.read_csv(filename, encoding='cp1250', delimiter=";")
812
813         if not df.empty:
814             mjesta = df['Mjesto'].unique()
815
816             df['Mjesto'] = df['Mjesto'].replace({mjesto: mjesta[i] for i, mjesto in enumerate(mjesta)})
817
818
```

Slika 48. Prikaz razlike funkcije testiranje i testiranje-pojedinačno.

Izvor: Autor

```

897     table_html = new_df_2024.to_html(index=False)
898
899
900     output += f"Predikcija za 2024 sa svim parametrima:\n\n"
901     output += table_html
902
903
904     self.text_edit.setGeometry(25, 50, 1500, 1000)
905     self.text_edit.setHtml(output)
906     self.text_edit.setStyleSheet("font-size: 12pt;")
907

```

Slika 49. Prikaz druge razlike funkcije testiranje-testiranje-pojedinačno.

Izvor: Autor

Predikcija vremena

Srednja kvadratna greška: 43.86009934010189 R²: 0.44540955733355186 Predikcija za 2024 sa svim parametrima:

Trenutačna Temperatura	Oborine	Vlaga	Vjetar	Trenutnovrijeme	Najvećatemperatura	Najmanjatemperatura	Datum	Mjesto	Dan
8.0	0.0	38.0	19.0	Poluoblačno	9.0	-3.0	2024-04-01	Zagreb	srijeda
4.0	1.0	57.0	7.0	Poluoblačno	10.0	-2.0	2024-04-02	Zagreb	četvrtak
7.0	77.0	70.0	8.0	Oblačno	13.0	3.0	2024-04-03	Zagreb	petak
9.0	85.0	88.0	4.0	Kiša	9.0	4.0	2024-04-04	Zagreb	subota
5.0	5.0	92.0	6.0	Vedro s povremenom naoblakom	8.0	-0.0	2024-04-05	Zagreb	nedjelja
15.0	1.0	35.0	7.0	Uglavnom sunčano	17.0	0.0	2024-04-06	Zagreb	ponedjeljak
8.0	8.0	66.0	5.0	Uglavnom sunčano	14.0	-1.0	2024-04-07	Zagreb	utorak
7.0	18.0	31.0	5.0	Oblačno	10.0	2.0	2024-04-08	Zagreb	srijeda
7.0	20.0	32.0	5.0	Poluoblačno	11.0	2.0	2024-04-09	Zagreb	četvrtak
5.0	88.0	20.0	7.0	Poluoblačno	9.0	7.0	2024-04-10	Zagreb	petak
9.0	73.0	8.0	8.0	Poluoblačno	11.0	5.0	2024-04-11	Zagreb	subota
10.0	82.0	14.0	14.0	Poluoblačno	11.0	7.0	2024-04-12	Zagreb	nedjelja
15.0	73.0	16.0	15.0	Poluoblačno	16.0	7.0	2024-04-13	Zagreb	ponedjeljak
13.0	85.0	16.0	16.0	Poluoblačno	13.0	7.0	2024-04-14	Zagreb	utorak
14.0	69.0	13.0	13.0	Poluoblačno	14.0	4.0	2024-04-15	Zagreb	srijeda
12.0	72.0	6.0	6.0	Poluoblačno	18.0	6.0	2024-04-16	Zagreb	četvrtak
12.0	66.0	5.0	5.0	Poluoblačno	18.0	5.0	2024-04-17	Zagreb	petak
13.0	66.0	5.0	5.0	Poluoblačno	21.0	7.0	2024-04-18	Zagreb	subota
17.0	57.0	8.0	8.0	Poluoblačno	22.0	11.0	2024-04-19	Zagreb	nedjelja
15.0	72.0	11.0	11.0	Poluoblačno	16.0	8.0	2024-04-20	Zagreb	ponedjeljak
13.0	68.0	8.0	8.0	Poluoblačno	17.0	6.0	2024-04-21	Zagreb	utorak
13.0	51.0	10.0	10.0	Poluoblačno	14.0	5.0	2024-04-22	Zagreb	srijeda
8.0	74.0	3.0	3.0	Poluoblačno	16.0	5.0	2024-04-23	Zagreb	četvrtak
13.0	80.0	5.0	6.0	Poluoblačno	19.0	7.0	2024-04-24	Zagreb	petak
17.0	56.0	7.0	8.0	Poluoblačno	23.0	11.0	2024-04-25	Zagreb	subota
17.0	51.0	18.0	18.0	Poluoblačno	18.0	7.0	2024-04-26	Zagreb	nedjelja
19.0	37.0	19.0	19.0	Poluoblačno	20.0	10.0	2024-04-27	Zagreb	ponedjeljak
16.0	70.0	10.0	10.0	Poluoblačno	18.0	11.0	2024-04-28	Zagreb	utorak
13.0	85.0	8.0	9.0	Poluoblačno	16.0	7.0	2024-04-29	Zagreb	srijeda

Slika 50. Prikaz predviđenih vrijednosti za 2024 u text editoru.

Izvor: Autor

6.4.1. Treniranje- testiranje-pojedinačno sučelje funkcije

Kad korisnik klikne na gumb „Unesi datoteku za treniranje i testiranje“, otvara mu se dijaloški prozor za odabir excel datoteke i implementira se funkcija import_files.

Nakon što je odabrana excel datoteka, ako je sve ispravno, izvršit će se treniranje i testiranje, podaci će se prikazati u tekstualnom editoru na sučelju.

6.5. Funkcije sučelja

Funkcije show_interface služe za prikazivanje određenog sučelja, autor je u aplikaciji koristio sveukupno 6 sučelja. Ako su podaci potrebni za sučelje prazni, prikazuje se upozorenje.

Ako već postoji trenutno prikazano sučelje, skriva se i zatim se stvara nova instanca sučelja i prikazuje se na određenoj geometriji ono sučelje koje je korisnik odabrao.

```
239 def show_interface1(self):
240     if df.empty or df_varazdin.empty or df_koprivnica.empty:
241         QMessageBox.warning(self, "Upozorenje", "Molimo odaberite excel datoteke kako bi ste vidjeli Statistiku.")
242         return
243     if self.current_interface:
244         self.current_interface.hide()
245         self.current_interface = None
246
247     self.current_interface = StatistikaApp(self)
248     self.current_interface.setGeometry(300, 50, 400, 400)
249     self.current_interface.show()
250
251
252 def show_interface2(self):
253     if df.empty or df_varazdin.empty or df_koprivnica.empty:
254         QMessageBox.warning(self, "Upozorenje", "Molimo odaberite excel datoteke kako bi ste vidjeli Vrijeme.")
255         return
256     if self.current_interface:
257         self.current_interface.hide()
258         self.current_interface = None
259
260     self.current_interface = PrvoSučelje(df, new_df_2024, places, self)
261     self.current_interface.setGeometry(300, 50, 400, 400)
262     self.current_interface.show()
263
264     pass
265
266 def show_interface3(self):
267     if df.empty or df_varazdin.empty or df_koprivnica.empty:
268         QMessageBox.warning(self, "Upozorenje", "odaberite excel datoteke kako bi ste vidjeli Prognozu vremena.")
269         return
270     if self.current_interface:
271         self.current_interface.hide()
272         self.current_interface = None
273
274     self.current_interface = third_interface()
275     pass
```

Slika 51. Prikaz koda funkcije show_interface()

Izvor: Autor

7.ZAKLJUČAK

Vrlo je bitno istaknuti značaj umjetne inteligencije (UI), neuronskih mreža, strojnog učenja i web scraping-a u području prognoze vremena, ali i svim ostalim područjima. Umjetna inteligencija i neuronske mreže omogućuju nam da se učinkovito obrade ogromne količine vremenskih podataka. Kroz trening neuronskih mreža i uz pomoć strojnog učenja, može se stvoriti model koji se prilagođava promjenama u vremenskim uvjetima i kontinuirano poboljšava svoje predikcijske sposobnosti.

Strojno učenje ima ključnu ulogu da „nauči“ iz podataka i primijeni ta znanja u predviđanju budućih vremenskih uvjeta. Web scraping nam omogućuje automatsko prikupljanje podataka s različitih internetskih izvora koji su bitni za analizu.

Važnost ovih tehnologija u području prognoze vremena je višestruka. Preciznija predviđanja vremenskih uvjeta omogućuju bolje planiranje i upravljanje aktivnostima koje su osjetljive na vremenske uvjete. Može imati i značajan utjecaj na različite sektore, kao što su poljoprivreda, turizam i mnogi drugi.

Kroz ovaj završni rad stekli smo dublje razumijevanje procesa prognoze vremena, prikupljanja i obrade podataka, razvoja korisničkog sučelja. Umjetna inteligencija, neuronske mreže, strojno učenje i web scraping predstavljaju ključne tehnologije u prognozi vremena. U budućnosti se očekuje daljnji napredak, što će rezultirati još preciznijom prognozom.

8. Literatura

[1.] Zyte, What is Web Scraping?[Online]

Dostupno na: <https://www.zyte.com/learn/what-is-web-scraping/> (3. 6.2023.)

[2.] Mathworks, What is a Neural Network?[Online].

Dostupno na: <https://www.mathworks.com/discovery/neural-network.html>

(3.6.2023.)

[3.]Dominodatalab, What is Anaconda?[Online]

Dostupno na: <https://www.dominodatalab.com/data-science-dictionary/anaconda> (4.6.2023.)

[4.] Teradata, What is python?[Online]

Dostupno na: <https://www.teradata.com/Glossary/What-is-Python> (4. 6. 2023.)

[5.] Baeldung, Multi-Layer Perceptron vs. Deep Neural Network[Online]

Dostupno na: <https://www.baeldung.com/cs/mlp-vs-dnn> (3. 6. 2023.)

[6.]Odsc, Why You Should be Using Jupyter Notebooks [Online]

Dostupno na: <https://odsc.medium.com/why-you-should-be-using-jupyter-notebooks-ea2e568c59f2> (5. 6. 2023.)

[7] Real Python, Jupyter Notebook: An Introduction[Online]

Dostupno na: <https://realpython.com/jupyter-notebook-introduction/> (5. 6. 2023.)

[8] JavaTpoint, Jupyter Notebook Tutorial[Online]

Dostupno na: <https://www.javatpoint.com/jupyter-notebook> (6. 6. 2023.)

[9] IBM, What is machine learning?[Online]

Dostupno na: <https://www.ibm.com/topics/machine-learning> (8. 6. 2023.)

[10] Spiceworks, What Is Machine Learning? Definition, Types, Applications, and Trends for 2022[Online]

Dostupno na: <https://www.spiceworks.com/tech/artificialintelligence/articles/what-is-ml/> (8. 6. 2023.)

[11]Investopedia, What Is a Neural Network?[Online]

Dostupno na: <https://www.investopedia.com/terms/n/neuralnetwork.asp> (9. 6. 2023.)

[12]Windy.app, What is a weather forecast and how does it work. Meaning, duration, accuracy, more[Online]

Dostupno na: <https://windy.app/blog/what-is-a-weather-forecast-and-how-it-works.html> [9. 6. 2023.]

[13] Medium, Machine Learning in Weather Forecasting[Online]

Dostupno na: <https://medium.com/meteum/machine-learning-in-weather-forecasting->

[fe6ad5b1e8e9](#) (9. 6. 2023.)

[14] Scaler, What is Pandas in Python?[Online]

Dostupno na: <https://www.scaler.com/topics/pandas/what-is-pandas-in-python/>(Citirano: 21. 6. 2023.)

[15] JavaTpoint, Python Pandas Tutorial[Online]

Dostupno na: <https://www.javatpoint.com/python-pandas> (21. 6. 2023.)

[16]Mygreatlearning, Python NumPy Tutorial – 2023[Online]

Dostupno na: <https://www.mygreatlearning.com/blog/python-numpy-tutorial/> (21. 6. 2023.)

[17] Tutorailspoint, What is Matplotlib? [Online]

Dostupno na: https://www.tutorialspoint.com/python_data_science/python_matplotlib.htm
(21. 6. 2023.)

[18] Tutorailspoint, What is Scikit-Learn (Sklearn)[Online]

Dostupno na: https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm
(21. 6. 2023.)

[19]Geegsforggeeks, Introduction to Tkinter[Online]

Dostupno na: <https://www.geeksforgeeks.org/introduction-to-tkinter/>
(21. 6. 2023.)

[20] Geegsforggeeks, Python datetime module[Online]

Dostupno na: <https://www.geeksforgeeks.org/python-datetime-module/> (21. 6. 2023.)

[21]JavaTpoint, PyQt library in Python[Online]

Dostupno na: <https://www.javatpoint.com/pyqt-library-in-python> (21. 6. 2023.)

[22] Simplilearn, Python Requests: Here's Everything You Should Know[Online]

Dostupno na: <https://www.simplilearn.com/tutorials/python-tutorial/python-requests>
(25.6.2023.)

[23] Crummy, Beautiful Soup Documentation[Online]

Dostupno na: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> (25.6.2023.)

[24] docs.python, CSV File Reading and Writing [Online]

Dostupno na: <https://docs.python.org/3/library/csv.html> (25.6.2023.)

9. Popis slika

Slika 1. Prikaz procesa web scrapinga.....	7
Slika 2. Struktura MLP-a.....	10
Slika 3. Prikaz početne strane anaconda navigator.	11
Slika 4. Prikaz sučelja jupyter notebook-a.....	12
Slika 5. Prikaz korištenih biblioteka u aplikaciji	14
Slika 6. Prikaz automatskog dohvaćanja podataka.	18
Slika 7. Identificirani HTML element.	18
Slika 8. Prikaz dohvaćenih podataka u konzoli.	19
Slika 9. Prikaz koda za unos podataka u excel.....	19
Slika 10. Prikaz excel-a s podacima.	20
Slika 11. Prikaz koda za glavno sučelje.....	22
Slika 12. Prikaz glavnog sučelja.	23
Slika 13. Prikaz sučelja Treniranje i testiranje.....	24
Slika 14. Prikaz koda sučelja Treniranje i testiranje.	25
Slika 15. Prikaz koda funkcija za odabir i učitavanje podataka.....	25
Slika 16. Prikaz koda za provjeravanje i pripremu podataka.....	26
Slika 17. Prikaz poruke s greškom	26
Slika 18. Prikaz koda za zamjenu tekstualnih vrijednosti.	26
Slika 19. Prikaz koda podijele podataka na skupove i kreiranje modela.	27
Slika 20. Prikaz koda predviđenih izlaznih vrijednosti i spremanja.	28
Slika 21. Prikaz koda za prikazivanje poruke.....	28
Slika 22. Prikaz uspješne poruke.....	29
Slika 23. Prikaz koda sučelja Statistika	30
Slika 24. Prikaz sučelja Statistika	31
Slika 25. Prikaz koda funkcije r2_mse	31
Slika 26. Prikaz R ² i MSE	31
Slika 27. Prikaz koda funkcije prikazi_statistiku_2023	32
Slika 28. Prikaz statistike za 2023.godinu.	33
Slika 29. Prikaz dijela koda s promijenjenim DataFrame za 2024.godinu.....	33
Slika 30. Prikaz statistike za 2024. godinu.	34
Slika 31. Prikaz koda sučelja Vrijeme.....	35
Slika 32. Prikaz dijela koda funkcije show_weather.....	36

Slika 33. Prikaz koda funkcije <code>show_next_place</code>	36
Slika 34. Prikaz sučelja vrijeme za mjesto Čakovec.	37
Slika 35. Prikaz koda Vremenska prognoza sučelja.....	38
Slika 36. Prikaz Vremenska prognoza sučelja.	39
Slika 37. Prikaz koda funkcije <code>stvar_pred_graph()</code>	40
Slika 38. Prikaz koda funkcije <code>prikazi_grafove()</code>	41
Slika 39. Prikaz koda funkcije <code>otvori_graf(mjesto)</code>	41
Slika 40. Prikaz grafa s parametrom najveća temperatura za mjesto Čakovec.....	42
Slika 41. Prikaz koda funkcije <code>prikazi_trenutno_vrijeme()</code>	43
Slika 42. Prikaz koda funkcije <code>otvori_graff()</code>	43
Slika 43. Prikaz grafa usporedba 2023 i 2024 za mjesto Varaždin.	44
Slika 44. Prikaz koda funkcije <code>prikazi_totalne_oborine</code>	44
Slika 45. Prikaz koda funkcije <code>prikazi_treci_graf()</code>	45
Slika 46. Prikaz koda za stvaranje i prikazivanje gumba na sučelju.	45
Slika 47. Prikaz sučelja treniranje-testiranje-pojedinačno.....	46
Slika 48. Prikaz razlike funkcije testiranje i testiranje-pojedinačno.	46
Slika 49. Prikaz druge razlike funkcije testiranje-testiranje-pojedinačno.	47
Slika 50. Prikaz predviđenih vrijednosti za 2024 u text editoru.	47
Slika 51. Prikaz koda funkcije <code>show_interface()</code>	48