



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

SKYCAST - APLICAȚIE DE PREDICȚIE METEO FOLOSIND INTELIGENȚĂ ARTIFICIALĂ

LUCRARE DE DISERTAȚIE

Absolvent masterand: **Ioan-Octavian STANCIU**

Coordonator **S.L. Dr. Ing Mădălin-Ioan NEAGU**
științific:

Iulie 2025

Cuprins

| | |
|--|-----------|
| Capitolul 1. Introducere | 1 |
| Capitolul 2. Obiectivele cercetării..... | 3 |
| 2.1. Obiectivul principal | 3 |
| 2.2. Obiective specifice..... | 3 |
| Capitolul 3. Studiu Bibliografic | 4 |
| 3.1. Aplicații mobile pentru oferirea informațiilor meteorologice | 4 |
| 3.2. Inteligența artificială folosită pentru predicții meteorologice | 5 |
| 3.3. Asemănări și deosebiri între diferite aplicații meteorologice | 6 |
| 3.3.1. Aplicația Apple Weather | 6 |
| 3.3.2. Aplicația Google Weather | 6 |
| 3.3.3. Aplicația The Weather Channel | 6 |
| 3.3.4. Aplicația AccuWeather..... | 7 |
| 3.3.5. Comparatie între aplicații meteo..... | 7 |
| 3.4. Avantajele platformei iOS comparativ cu Android | 8 |
| 3.5. Tehnologii folosite..... | 9 |
| Capitolul 4. Prezentarea proiectului..... | 11 |
| 4.1. Cerințe funcționale..... | 11 |
| 4.2. Cerințe non-funcționale | 13 |
| 4.3. Prezentarea regresiiilor utilizate | 14 |
| 4.3.1. Regresie liniară | 14 |
| 4.3.2. Regresie „Random Forest” | 15 |
| 4.3.3. Regresie „Boosting Tree”..... | 16 |
| 4.3.4. Regresie „Decision Tree” | 17 |
| 4.4. Metricile de evaluare utilizate | 17 |
| 4.4.1. Eroarea medie absolută (MAE) | 17 |
| 4.4.2. Eroarea medie pătratică (MSE) | 18 |
| 4.4.3. Root Mean Squared Error..... | 18 |
| 4.4.4. Coeficientul de determinare..... | 18 |
| 4.5. Structura proiectului | 18 |
| 4.5.1. Structura fișierelor | 18 |
| 4.5.2. Structura interfeței grafice | 19 |
| 4.6. Structura model de date | 21 |
| 4.6.1. Structura modelului de date OpenMeteo | 21 |
| 4.6.2. Structura modelului de date WeatherAPI..... | 22 |

| | |
|--|-----------|
| 4.6.3. Structura modelului de date SkyCast..... | 23 |
| 4.7. Antrenarea modelului | 24 |
| 4.7.1. Crearea și antrenarea modelului folosind Create ML..... | 24 |
| 4.7.2. Crearea modelului în mod programatic | 27 |
| 4.7.3. Antrenarea modelului în modul programatic..... | 28 |
| 4.7.4. Afișarea datelor..... | 30 |
| Capitolul 5. Rezultate teoretice și experimentale | 32 |
| 5.1. Rezultate folosind Regresie liniară..... | 32 |
| 5.2. Rezultate folosind Regresie Random Forest..... | 37 |
| 5.3. Rezultate folosind Regresie Boosted Tree..... | 39 |
| 5.4. Rezultate folosind Regresie Decision Tree..... | 42 |
| 5.5. Comparatie între regresorii disponibili..... | 44 |
| 5.6. Comparatie între SkyCast și alte aplicații meteorologice..... | 47 |
| Capitolul 6. Concluzii..... | 50 |
| 6.1. Contribuții proprii | 51 |
| 6.2. Dezvoltări ulterioare | 51 |
| Bibliografie..... | 53 |
| Anexa 1. Exemple de cod | 54 |
| Anexa 2 Lista figurilor și a tabelelor | 56 |

Capitolul 1. Introducere

Acest proiect își propune dezvoltarea și implementarea unei aplicații mobile de prezicere a vremii din ziua curentă și a următoare zile pentru a oferi utilizatorilor anumite metrice cât mai precise și să ajute la evitarea cazurilor nefericite în care oamenii au parte de o vreme diferită față de cum se așteptau în urma verificării altor aplicații dedicate aceluiași scop.

Odată cu revoluționarea inteligenței artificiale prin lansarea platformei ChatGPT de către compania OpenAI în anul 2022 [1], lumea a început să se miște într-un ritm extrem de alert încercând pe cât mai mult posibil automatizarea lucrurilor de zi cu zi folosind asemenea tehnologii. Atât companii specializate în domeniul IT cât și companii ce produc electrocasnice inteligente sau chiar cele producătoare de vehicule. IA-ul ajunge ușor ușor omniprezent în viețile tuturor, iar persoanele de toate vârstele încep să nu mai aprecieze așa mult valoarea acestuia și să fie reticente în încercarea unor noi experiențe dacă acestea includ inteligența artificială. De asemenea, cea mai întâlnită problemă când vine vorba de diferite obiecte ce folosesc AI este confidentialitatea datelor. Deși multe companii folosesc AI, acestea au fost destul de reticente la început deoarece nu știau dacă informații despre dezvoltarea produselor lor pot fi dezvaluite în urma folosirii unor astfel de instrumente.

Oamenilor li se întâmplă foarte frecvent să verifice vremea pentru următoarea zi, iar vremea de afară nu corespunde cu cea indicată în aplicație. Acest lucru poate să ajungă foarte frustrant și supărat în special în cazurile în care anumite acțiuni depind de acest lucru. Cineva care vrea să ajungă la un interviu foarte important, nu ar vrea să plece de acasă fără o umbrelă în cazul unei ploii. Alții își fac bagajul pentru vacanță și vor să știe cele mai importante lucruri pe care să le poată lua cu ei.

În plus, nefiind sigur că o singură aplicație îți arată foarte precis vremea ce va fi în următoarea zi sau chiar în ziua curentă, telefoanele noastre ajung să aibă minim două aplicații pentru a ajuta la această problemă, iar acest lucru poate să fie foarte derutant pentru un utilizator. Unui utilizator îi trebuie o singură aplicație care să fie cât mai precisă, cu date cât mai de actualitate și să reducă rata de eroare până aproape de zero. Desigur, acest lucru este imposibil din moment ce acestea depinde de un factor care este imprevizibil și anume: vremea.

Ce poate să ajute la remedierea acestei probleme este, desigur, și citirea meteo făcută de stații specializate în acest domeniu și chiar urmărirea știrilor. Cu toate acestea, trăim într-un secol în care fiecare dintre noi avem în buzunar un mini-calculator pe care îl putem folosi la capturarea video, foto, adăugări de notițe și comunicarea cu persoanele apropiate, oriunde s-ar afla acestea pe Pământ. Pe lângă aceste avantaje, smartphone-urile conțin o gramadă de informații ce pot fi accesate prin intermediul internetului. Totuși preferăm în ziua de astăzi să dea o căutare pe internet pentru a ne informa corect în legătură cu orice nelămurire are, iar asta include și informarea în legătură cu vremea pentru următoarea zi și ziua curentă.

O altă problemă care poate să apară este o schimbare bruscă de vreme fără ca utilizatorii să fie conștienți. Ce poate ajuta la îmbunătățirea acestui lucru sunt notificări sau chiar mesaje trimise utilizatorilor pentru a fi înștiințați de astfel de schimbări și pentru a se putea pregăti cât mai mult și repede pentru ele.

Cu toate acestea, aplicația SkyCast ajută la remedierea acestor probleme și oferă utilizatorilor o prezicere cât mai precisă a vremii folosind inteligența artificială fără ca folosirea unor date legate de utilizatori, toate în același sistem. Aceasta are o interfață simplă,

intuitiva ce contine doar informatii necesare legate de vreme fara a fii prea ingramadita de alte lucruri ce ar putea sa nu fie de interes pentru utilizatori.

SkyCast ofera clientilor posibilitatea de a putea viziona vremea si de a stii cu exacitate cate grade celsius o sa fie, cum o sa se simta acele grade si procentul de sanse de precipitatii pentru urmatoarele 24 de ore de cand acestia intra in aplicatie. Aceasta colecteaza asemenea informatii din diferite surse, iar cu ajutorul inteligentei artificiale se obtine o prezicere cat mai buna pentru a asigura un nivel de încredere cat mai mare pentru fiecare utilizator.

Capitolul 2. Obiectivele cercetării

2.1. Obiectivul principal

Obiectivul principal al acestui proiect e de a dezvolta o aplicație mobilă iOS care are rolul de a oferi utilizatorilor diferite informații, cât mai precise cu vedere la temperatura actuală și probabilitatea de precipitații atât din ziua curentă cât și din cea următoare, iar pe lângă aceasta să se realizeze o comparație între algoritmi folosiți la predicția acestor două lucruri pentru a stabili care este cel mai potrivit algoritm de antrenare al acestor date. La finalul acesteia, se va realiza o comparație între datele prezise de către aplicație și cele dispuse de către Apple Weather și Google Weather.

Deși ne aflăm în „era tehnologiei”, niciodată nu a existat o aplicație care să ofere o acuratețe de sută la sută când vine vorba de prezicere a vremii, indiferent de ce se dorește a afla (probabilitatea de precipitații, viteza vântului, indicele UV, procentajul de umiditate etc). Fiecare persoană trebuie să se informeze din cel puțin două surse diferite pentru a își putea face o idee despre cum o să fie vremea în locul pe care dorește să îl viziteze și pentru a-și putea lua măsuri de precauție în cazul unor fenomene meteorologice.

2.2. Obiective specifice

Posibilitatea de a obține date din diferite surse: Aplicația permite utilizatorilor să vizualizeze informații despre temperatură și probabilitatea de precipitații din ultimele șapte zile din două surse diferite: Open-meteo[2] și WeatherAPI[3].

Posibilitatea de a putea alege algoritmul de antrenare: Utilizatorul poate selecta unul dintre cei patru algoritmi ce pot fi folosiți la antrenarea modelului cu datele preluate de pe cele două site-uri precizate anterior: regresie liniară, regresie "random forest", regresie "boosting tree" și regresie cu arbori de decizie.

Metrici de evaluare: Sunt folosite patru metrici de evaluare a performanței modelului antrenat care sunt afișate în partea de sus a interfeței: eroarea medie absolută, eroare pătratică medie, rădăcină pătrată a erorii pătratice a mediei și coeficientul de determinare.

Posibilitatea de a putea alege ziua: Utilizatorul are opțiunea de a alege dacă dorește să vadă informațiile meteorologice pentru ziua curentă sau pentru ziua următoare. Această posibilitate îi permite să își facă o idee mai bună a vremii pentru a putea lua cea mai bună alegere în privința planurilor pe care dorește să și le facă.

Afișarea informațiilor pentru fiecare oră: Aplicația afișează sub formă de listă, pentru ziua selectată, informațiile meteorologice pentru fiecare oră a zilei selectate. Acest lucru permite formarea unei facilitează înțelegerea mai bună a vremii.

Afișarea informațiilor sub formă de diagrame: Utilizatorul poate selecta opțiunea de vizualizare a datelor sub forma unei diagrame a cărei axă X reprezintă toate orele dintr-o zi și axa Y reprezintă variațiile valorilor selectate.

Compararea rezultatelor: La finalul lucrării, se va efectua o comparație între acuratețea aplicației raportându-ne la informații oferite de alte site-uri/aplicații din același domeniu.

Capitolul 3. Studiu Bibliografic

3.1. Aplicații mobile pentru oferirea informațiilor meteorologice

Aplicațiile de informare meteorologică reprezintă o parte esențială din viața fiecărei persoane deținătoare de un smartphone, oferind utilizatorilor acces rapid la datele meteorologice, indiferent de locație. Odată cu integrarea acestora în diferite platforme, aplicațiile de acest tip sunt din ce în ce mai populare crescând nevoia de planificare în funcție de condițiile meteo. Platformele de prognoză meteo precum Apple Weather și Google Weather au revoluționat modul în care oamenii accesează informațiile meteorologice și oferă date în timp real despre temperatură, umiditate, precipitații, vânt și alți parametri relevanți.

Nu numai că aceste aplicații oferă informații în timp real, ci oferă și integrări cu alte aplicații mobile precum calendarele sau aplicațiile de fitness cu scopul de a oferi utilizatorilor prognoze pe termen scurt și lung, alerte meteo personalizate. Informațiile sunt afișate într-un format vizual intuitiv și atractiv care este accesibil atât utilizatorilor obișnuiți, cât și experților în domenii precum agricultură, transport, turism sau organizarea de evenimente în aer liber.

Astăzi, aplicațiile meteo joacă un rol important în luarea deciziilor zilnice, de la alegerea îmbrăcăminte potrivite, până la planificarea vacanțelor sau evenimentelor. Spre exemplu, înainte de a pleca în excursii sau în concedii, utilizatorii pot consulta aplicațiile pentru a evita perioadele cu vreme în defavoarea lor. În plus, aceste aplicații sunt folosite intens în industrii care depind de vreme, cum ar fi aviația, construcțiile sau pescuitul.

Conform unui raport publicat de Grand View Research [4], piața globală a aplicațiilor meteo este estimată să depășească valoarea de 1 miliard de dolari în următorii ani, ca urmare a creșterii numărului de utilizatori și a îmbunătățirii continue a tehnologiilor de predicție meteorologică. Această creștere este susținută și de integrarea algoritmilor de inteligență artificială, care permit generarea unor prognoze mai precise și personalizate pentru fiecare utilizator.

Printre cele mai utilizate aplicații de vreme la nivel global se numără:

1. **Apple Weather** – aplicație preinstalată pe dispozitivele iOS, integrată cu Siri și cu funcții avansate de prognoză în timp real.
2. **Google Weather** – disponibilă prin Google Search și Google Assistant, oferă informații rapide și actualizate despre condițiile meteorologice.
3. **The Weather Channel** – una dintre cele mai populare aplicații independente, cunoscută pentru acuratețea prognozelor și alertele meteo hiper-locale.

Pe lângă aplicațiile oferite de unele dintre cele mai de succes companii, Apple și Google, există și numeroase aplicații de vreme dezvoltate de companii specializate în domeniul meteorologic. De exemplu, The Weather Channel [5] și Weather Underground [6] oferă informații detaliate și personalizate, incluzând radar meteo, predicții pe termen lung, hărți interactive și analize climatice. Unele aplicații permit chiar selectarea mai multor locații favorite, trimiterea de notificări în caz de fenomene extreme sau oferirea de recomandări pentru activități zilnice, cum ar fi alergatul. Utilizatorii pot astfel lua decizii bine informate în funcție de prognozele actualizate constant, optimizându-și rutina zilnică sau activitatea profesională.

Aplicațiile meteo moderne devin tot mai interactive și integrate în ecosisteme digitale mai largi. Un exemplu ar fi posibilitatea conectării cu dispozitive smart home, cum ar fi termostatele inteligente, care își ajustează automat setările în funcție de temperaturile prognozate. CARROT Weather [7], adaugă un element de divertisment prin oferirea prognozelor într-un stil umoristic și personalizabil. În domeniul agriculturii de precizie,

aplicațiile meteo avansate colaborează cu senzori de sol și drone pentru a oferi fermierilor date esențiale privind momentul optim pentru irigații sau recoltare. Astfel, aplicațiile meteorologice nu doar că răspund nevoilor individuale, dar contribuie semnificativ și la eficiența operațională în diverse domenii industriale.

3.2. Inteligența artificială folosită pentru predicții meteorologice

Inteligența artificială a devenit un instrument esențial în modernizarea metodelor de prognoză meteorologică, în special într-un context marcat de creșterea cantității de date climatice colectate în timp real. În trecut, modelele numerice de predicție a vremii (NWP) erau principalul standard pentru estimarea parametrilor atmosferici, însă aceste metode au limitări legate de complexitatea computațională și de sensibilitatea la erori de condiții inițiale. IA, prin abordări bazate pe învățare automată (ML) și învățare profundă (DL), propune alternative rapide, adaptabile și capabile să exploateze tipare ascunse în volume mari de date istorice și în timp real. [8]

Unul dintre principalele avantaje ale AI este capacitatea sa de a învăța din date fără a necesita o modelare explicită a tuturor fenomenelor fizice. Modelele de învățare automată pot identifica relații complexe neliniare între variabile meteorologice, lucru dificil de realizat prin metode clasice. Astfel, predicțiile devin mai precise pentru parametri precum temperatura, umiditatea, viteza vântului sau probabilitatea de precipitații.

Pe măsura ce senzorii de pe sateliți, stații meteo și rețele IoT generează volume tot mai mari de date, inteligența artificială poate acționa ca o interfață eficientă de extragere a informației utile. Modele precum rețelele neuronale convoluționale (CNN), rețele recurente (RNN, LSTM) sau arhitecturile hibride sunt utilizate pentru a anticipa evoluția fronturilor atmosferice sau pentru a detecta evenimente extreme. [10]

Implementarea acestor modele necesită o pregătire riguroasă a datelor. Datele meteorologice sunt afectate de sezonabilitate, zgomot și lipsuri. IA-ul necesită tehnici robuste de preprocesare: interpolare, normalizare, eliminarea valorilor aberante și conversia datelor temporale în reprezentări compatibile cu modelele. În acest context, abordările de tip „data fusion”, care combină surse diverse (de la observații la sol, date satelitare și modele numerice) contribuie la creșterea performanței predicțiilor.

Un exemplu practic este utilizarea IA-ului pentru predicția temperaturii într-un oras european, folosind date de la stații locale și modele climatice globale. Modelele ML pot fi antrenate să înțeleagă influența unor factori locali precum tipul solului, densitatea urbană sau istoricul termic recent. Rezultatul este o predicție granulară, presonalizată pentru regiuni mici, care poate fi recalculată în timp real în funcție de noile date disponibile.

În plus, IA-ul a început să fie folosită în detectarea timpurie a fenomenelor extreme, cum ar fi valurile de căldură, furtunile violente sau inundațiile rapide. Modelele pot învăța din evenimente trecute și pot semnaliza apariția unui tipar asemănător, oferind astfel autorităților locale sau populației avertizări cu anticipație utilă. Această funcționalitate este crucială într-un context marcat de schimbări climatice tot mai pronunțate. [12]

Cu toate acestea, există provocări în utilizarea AI pentru predicții meteo. Modelele pot învăța relații eronate dacă datele sunt dezechilibrate sau dacă prezintă corelații false. De asemenea, lipsa transparenței în modul de funcționare al unor rețele neuronale complică validarea științifică. Din acest motiv, cercetătorii explorează metode de interpretabilitate, cum ar fi LIME sau SHAP, care explică contribuția fiecărui factor la predicția finală.

Un alt aspect discutat în literatura de specialitate este necesitatea calibrării periodice a modelelor AI. Condițiile meteorologice se pot schimba de la un sezon la altul, iar modelele antrenate pe date istorice trebuie să se adapteze prin reantrenare continuă. Această capacitate este facilitată de infrastructura modernă de procesare cloud și de algoritmi care învață incremental.

Pe lângă partea științifică, AI are și o componentă aplicativă foarte importantă. Aplicațiile mobile care oferă prognoze meteo personalizate, sistemele de avertizare timpurie pentru sectorul agricol sau logistic și optimizarea energetică prin predicția cererii în funcție de condițiile atmosferice sunt exemple concrete de implementare practică.

3.3. Asemănări și deosebiri între diferite aplicații meteorologice

3.3.1. Aplicația Apple Weather

Apple Weather este aplicația nativă de informare meteorologică disponibilă pe toate dispozitivele vândute de compania Apple și este strâns integrată cu ecosistemul oferit de aceștia. Interfața aplicației este elegantă, intuitivă și minimalistă, oferind informații esențiale despre vreme în timp real: temperatură actuală, prognoză pe ore și pe zece zile, umiditate, probabilitate de precipitații, direcția și viteza vântului, indicele UV și faza lunii. Informațiile sunt prezentate într-o manieră vizuală rafinată, cu animații dinamice care reflecta condițiile meteo actuale.

Începând cu versiunea de iOS 15, Apple a integrat mai multe funcții preluate de la aplicația Dark Sky [9], pe care a achiziționat-o în anul 2020. Așadar, Apple Weather oferă acum prognoze hiper-locale, care anunță cu acuratețe când va începe sau se va opri ploaia în următoarele minute, în funcție de poziția utilizatorului. De asemenea, utilizatorii pot primi notificări privind fenomene meteorologice extreme, cum ar fi furtuni, valuri de căldură sau înghețuri.

Un avantaj important al aplicației este integrarea cu Siri și cu widget-urile din ecranul de pornire, ceea ce permite acces rapid la informații fără a deschide efectiv aplicația. Cu toate acestea, aplicația este disponibilă exclusiv pe dispozitivele deținute de Apple, limitând accesibilitatea pentru utilizatorii altor sisteme de operare. De asemenea, nivelul de detaliu al informațiilor poate varia în funcție de regiune, iar personalizarea aplicației este relativ limitată în comparație cu alte platforme.

3.3.2. Aplicația Google Weather

Google Weather este o aplicație implicită integrată în Google Search și Google Assistant, disponibilă pe dispozitive Android și accesibilă și prin browser. Interfața este simplă și axată pe lizibilitate, oferind date rapide despre vremea curentă, prognoza pe ore și pe mai multe zile, nivelul UV, viteza vântului și umiditatea. Aplicația este gândită pentru utilizare rapidă, fiind adesea accesată printr-o simplă întrebare adresată Asistenului Google, precum „Cum e vremea azi?”.

Unul dintre avantajele majore ale Google Weather este accesibilitatea multiplatformă. Poate fi folosită fără instalarea unei aplicații dedicate, funcționând eficient direct din motorul de căutare sau din interfața vocală. Mai mult, Google folosește o rețea vastă de surse meteorologice și algoritmi de învățare automată pentru a rafina prognoza meteo, adaptând-o în funcție de locația utilizatorilor.

Totuși, aplicația nu oferă o gamă la fel de largă de opțiuni vizuale sau personalizări precum alte aplicații de vreme. Lipsa unui radar meteo sau a hărților interactive poate limita utilitatea pentru utilizatorii care doresc informații mai tehnice sau o experiență vizuală mai bogată. Cu toate acestea, pentru majoritatea utilizatorilor care doresc informații esențiale într-un timp cât mai scurt, Google Weather rămâne o soluție eficientă și comodă.

3.3.3. Aplicația The Weather Channel

The Weather Channel este una dintre cele mai vechi și respectate aplicații meteo independente, disponibilă pe toate platformele majore (iOS, Android, web). Este cunoscută

pentru acuratețea prognozelor sale și pentru volumul mare de informații disponibile. Aplicația oferă prognoze pe termen scurt și lung, alerte personalizate, radar meteo interactiv, indice UV, calitatea aerului, vizibilitatea și temperatura resimțită, toate prezentate într-o interfață complexă, dar bine organizată.

Un studiu realizat de ForecastWatch a concluzionat că The Weather Channel oferă cele mai precise prognoze zilnice, depășind alte aplicații populare precum AccuWeather și Apple Weather. Aplicația permite configurarea de alerte personalizate în timp real și oferă o experiență ușor de utilizat, fără a compromite calitatea serviciului.

În comparație cu alte aplicații, The Weather Channel oferă un nivel ridicat de personalizare și o gamă largă de funcționalități, fiind apreciată de utilizatorii care doresc informații detaliate și fiabilitate în predicții.

3.3.4. Aplicația AccuWeather

AccuWeather este una dintre cele mai cunoscute aplicații meteo independente, apreciată pentru precizia prognozelor și pentru gama sa largă de funcționalități fiind valabilă pe platformele iOS, Android și web. Câteva caracteristici principale sunt prognozele pe termen atât scurt cât și lung, afișarea temperaturii resimțite, indicii UV, calitatea aerului, radar meteo și informații detaliate despre precipitații și vânt. Un element distinctiv este funcția „MinuteCast” care oferă prognoze la minut pentru următoarele două ore, adaptate la locația exactă a utilizatorului.

AccuWeather permite personalizarea alertelor meteo, salvarea mai multor locații și oferă o interfață complexă, dar ușor de navigat pentru utilizatorii familiarizați cu aplicațiile de acest tip. Se pare că publicul țintă al acesteia este cel creat din persoane care au nevoie de detalii tehnice precum piloții, sportivi sau fermieri. De asemenea, aplicația include grafice detaliate și hărți animate, utile pentru anticiparea mișcării furtunilor sau planificarea unor activități în aer liber.

Deși aplicația este gratuită, versiunea standard conține reclame, ceea ce poate afecta experiența utilizatorului. Totuși, există și o versiune premium, fără reclame, care oferă acces la informații suplimentare. Aceasta este recunoscută în mod constant pentru fiabilitatea sa, fiind adesea în topurile internaționale privind acuratețea prognozelor, alături de The Weather Channel.

3.3.5. Comparație între aplicații meteo

Apple Weather, Google Weather, The Weather Channel și AccuWeather sunt aplicații meteo populare care oferă utilizatorilor informații esențiale despre condițiile atmosferice actuale și prognozele viitoare. Toate cele patru aplicații afișează temperatura curentă, umiditatea, probabilitatea de precipitații și prognoze pe termen scurt și lung. De asemenea, ele oferă acces rapid la informații prin diverse metode: widget-uri pe ecranul principal (Apple Weather, Google Weather și AccuWeather), interfață vocală (GoogleWeather și Apple Weather) sau alerte personalizate (The Weather Channel și AccuWeather). Interfața este simplificată în cazul Apple și Google, în timp ce restul aplicațiilor pun accent pe o prezentare vizuală complexă, cu hărți radar interactive, graficele detaliate și prognoze pe minute.

Diferențele majore apar în ceea ce privește platforma, gradul de personalizare și acuratețea prognozelor. Apple Weather este disponibilă doar pentru utilizatorii iOS și se integrează perfect cu Siri și alte servicii Apple, dar oferă personalizare limitată. Google Weather, ușor accesibilă prin Google Search sau Google Assistant, este minimalistă și rapidă, dar cu funcții reduse. În schimb, The Weather Channel, este disponibilă pe toate platformele majore, oferă o gamă extinsă de funcții, inclusiv alerte detaliate, hărți radar și prognoze personalizate. La fel, AccuWeather oferă prognoze hipster-locale prin funcția MinuteCast,

opțiuni avansate de personalizare și este apreciată pentru acuratețea ridicată a predicțiilor sale, mai ales în rândul utilizatorilor profesioniști sau cu nevoi meteo specifice.

În concluzie, alegerea aplicației potrivite depinde de preferințele și nevoile fiecărui utilizator. Apple Weather este ideală pentru cei care caută o experiență elegantă și bine integrată cu iOS. Google Weather este potrivită pentru informare rapidă, fără instalarea unei aplicații dedicate. The Weather Channel răspunde cel mai bine celor care au nevoie de funcționalități avansate și alerte detaliate, iar AccuWeather este o alegere excelentă pentru utilizatorii care caută acuratețe, detalii tehnice și personalizare extinsă.

Tabel 3.1 - Comparație Apple Weather, Google Weather și The Weather Channel

| Criteriu | Apple Weather | Google Weather | The Weather Channel | AccuWeather |
|----------------------|----------------------------------|--------------------------------|---|------------------------------------|
| Platformă | iOS exclusiv | Android și Web | iOS, Android, Web | iOS, Android, Web |
| Interfață | Estetică, minimalistă | Simplă, clară | Complexă, dar detaliată | Detaliată, tehnică |
| Acuratețea prognozei | Ridică, mai ales local | Bună, rapidă | Foarte ridicată | Foarte ridicată (MinuteCast) |
| Funcții avansate | Hiper-localizare, integrare Siri | Răspuns vocal, ușor de accesat | Radar, alerte personalizate, hărți | MinuteCast, radar, grafice meteo |
| Personalizare | Limitată | Foarte redusă | Extinsă (inclusiv alerte personalizate) | Extinsă (alerte, locații multiple) |
| Publicitate | Fără | Fără | Da, în versiunea gratuită | Da, în versiunea gratuită |
| Public țintă | Utilizatori iOS | Utilizatori Android/Google | Utilizatori avansați, profesioniști | Utilizatori avansați, tehnici |

3.4. Avantajele platformei iOS comparativ cu Android

Platforma iOS, dezvoltată de Apple, conform unui articol de pe site-ul Lerator [11] oferă o experiență de utilizare considerată de mulți ca fiind mai fluidă și mai coerentă decât cea oferită de Android. Acest lucru se datorează faptului că Apple controlează atât hardware-ul, cât și software-ul dispozitivelor sale, ceea ce permite o integrare perfectă între sistemul de operare și comportamentele fizice. Această sinergie se reflectă într-o performanță optimizată pe termen lung, stabilitate în rularea aplicațiilor și un design unitar al interfeței, indiferent de modelul de iPhone folosit. Totodată, aplicațiile disponibile pe iOS sunt adesea mai bine finisate, dezvoltatorii concentrându-se pe această platformă datorită uniformității sistemului și a publicului dispus să plătească pentru conținut de calitate.

Un alt avantaj major al ecosistemului iOS este frecvența și rapiditatea actualizărilor de software. În timp ce utilizatorii Android depind de producători și operatori pentru a primi ultimele versiuni ale sistemului de operare, utilizatorii de iPhone beneficiază de actualizări simultane, direct de la Apple, pe o perioadă mai lungă de timp - chiar și 5-6 ani de suport. Această uniformitate reduce riscurile de vulnerabilități de securitate și menține dispozitivele actualizate din punct de vedere funcțional și estetic. De asemenea, sistemul iOS este recunoscut pentru politicile stricte de securitate, care oferă o protecție superioară împotriva malware-ului și accesului neautorizat la datele utilizatorului.

În plus, iOS oferă o integrare impecabilă cu restul dispozitivelor Apple, ceea ce creează un ecosistem unitar și eficient. Un utilizator de iPhone poate, de exemplu, să înceapă

scrierea unui mesaj pe telefon și să continue instantaneu pe MacBook, să răspundă la apeluri de pe Apple Watch sau să transfere fișiere prin AirDrop fără aplicații terțe. Această interoperabilitate aduce un plus de confort și viteză în utilizarea zilnică a tehnologiei. În același timp, dispozitivele Apple își păstrează valoarea în timp, fiind apreciate pe piața second-hand pentru durabilitate, suport software extins și design atemporal.

3.5. Tehnologii folosite

În dezvoltarea acestei aplicații, SkyCast, alegerea tehnologiilor utilizate a fost dictată nu doar de compatibilitatea cu platforma țintă, ci și de dorința de a obține un echilibru optim între performanță, flexibilitate, experiență de utilizare și ușurința de dezvoltare. Ecosistemul Apple oferă o integrare unitară între hardware și software, motiv pentru care întregul proces de dezvoltare a fost realizat exclusiv pe un MacBook, folosind Xcode ca mediu de dezvoltare principal. Acest tip de laptop, datorită sistemului său de operare macOS și a arhitecturii optimizate pentru sarcini de dezvoltare, permite rularea fluentă a aplicațiilor de programare, simulatoarelor și testelor, fără întârzieri semnificative sau blocaje. Procesorul performant, alături de memoriile rapide, a susținut antrenarea modelelor CreateML direct pe dispozitiv, fără a necesita resurse externe, ceea ce a oferit un avantaj important în testarea rapidă a performanței și ajustarea hiperparametrilor.

Platforma Xcode a fost esențială în întreg procesul de programare, având o interfață intuitivă și o suită completă de unelte pentru dezvoltarea aplicațiilor iOS. Editorul de cod oferă sugestii în timp real, suport pentru debugging eficient și simulatoare variate, ceea ce a permis testarea aplicației pe diverse modele de iPhone, fără a avea fizic acces la toate acestea. În plus, integrarea automată cu simulatorul iOS și cu instrumentele pentru monitorizarea consumului de memorie și procesor a oferit o imagine detaliată a performanței aplicației în timp real. Aceste unelte sunt extrem de utile, mai ales în cazul aplicațiilor care gestionează date complexe și actualizări frecvente, cum este cazul aplicațiilor meteo bazate pe modele de predicție.

Limbajul de programare folosit este Swift, un limbaj modern, dezvoltat de Apple, care oferă siguranță sporită la rulare și o sintaxă clară, adaptată pentru aplicații mobile. Spre deosebire de limbajele tradiționale, Swift reduce riscul de erori prin tipuri opționale și tratări sigure ale valorilor de tip nil, fiind astfel potrivit pentru aplicațiile care prelucrează date incerte sau incomplete, cum este cazul valorilor meteo. În plus, performanța sa este comparabilă cu a limbajelor compilate, permițând o execuție rapidă, aspect crucial în rularea modelelor de regresie pe dispozitiv.

SwiftUI, framework-ul declarativ pentru construirea interfeței grafice, a fost o alegere inspirată în proiect, întrucât permite o organizare clară a componentelor vizuale și o adaptare ușoară a acestora la diferite dimensiuni de ecran. Această tehnologie elimină nevoia gestionării manuale a constrângerilor și oferă o modalitate rapidă de actualizare a interfeței în funcție de schimbările de stare. În cazul aplicației SkyCast, unde predicțiile și datele meteo se actualizează la intervale regulate, această capacitate de reîmprospătare dinamică a fost esențială. Designul minimalist și adaptiv al aplicației a fost realizat tocmai pentru a oferi o experiență fluentă, neîncărcată, în care accentul cade pe datele utile pentru utilizator.

Pentru componenta de inteligență artificială și învățare automată, a fost utilizat CreateML, un instrument pus la dispoziție de Apple, ideal pentru dezvoltatorii care nu doresc să interacționeze direct cu biblioteci externe complexe precum TensorFlow sau PyTorch. CreateML permite crearea de modele de regresie, clasificare și clustering direct în Xcode, folosind interfața grafică sau cod Swift. În cadrul aplicației, au fost antrenate patru tipuri de regresori (Linear Regression, Random Forest, Boosted Tree, Decision Tree) pentru estimarea temperaturii, temperaturii resimțite și probabilității de precipitații. Modelele au fost testate pe

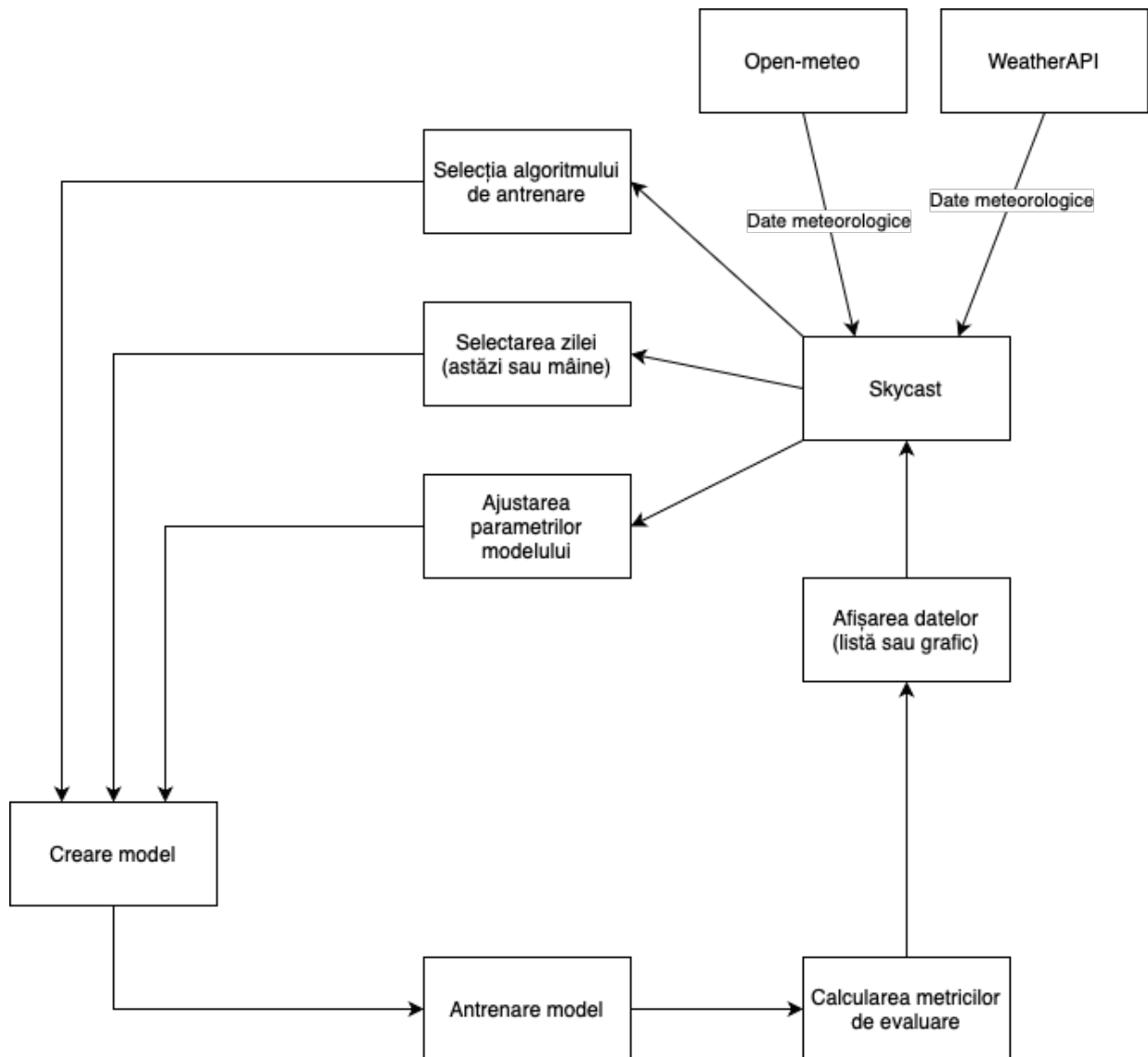
date reale, obținute din surse precum WeatherAPI și OpenMeteo, iar rezultatele au fost exportate și analizate cu ușurință datorită integrării direct în mediu.

Un alt beneficiu important al dezvoltării pe platforma Apple a fost reprezentat de integrarea Chart, o bibliotecă nativă ce permite afișarea de grafice personalizate și interactive. În contextul acestei aplicații, a fost utilizată pentru a vizualiza valorile reale și predicțiile fiecărui model, într-un mod comparativ. Acest lucru a fost deosebit de util pentru analiza performanței în timp și pentru ajustarea modelelor în funcție de erorile înregistrate. Vizualizările au fost construite astfel încât utilizatorul să poată observa cu ușurință abaterile dintre predicțiile modelelor și valorile reale furnizate de serviciile comerciale.

Pentru gestionarea codului sursă, a fost folosit Git, un sistem de control al versiunilor care a permis salvarea progresivă a proiectului, revenirea la versiuni anterioare și colaborarea eficientă între etapele de dezvoltare și testare. SourceTree a oferit o interfață vizuală pentru git, facilitând urmărirea modificărilor și evitarea conflictelor între diferite ramuri ale proiectului. Această abordare a permis documentarea corectă a etapelor de dezvoltare și menținerea unei structuri clare a codului, aspect esențial în cazul aplicațiilor cu mai multe componente și logici de predicție complexe.

În concluzie, întregul ecosistem tehnologic de la Apple, de la hardware la software și unelte auxiliare a fost ales pentru a susține o dezvoltare coerentă, eficientă și scalabilă. Alegerea acestor tehnologii a permis nu doar realizarea unei aplicații funcționale, ci și testarea sistematică a performanței fiecărui model de regresie într-un mod transparent și reproductibil. Astfel, aplicația SkyCast nu este doar un instrument de vizualizare a vremii, ci și o platformă de analiză și testare a modelelor de inteligență artificială, cu potențial de extindere în scenarii mai complexe.

Capitolul 4. Prezentarea proiectului



Figură 4.1 - Diagramă de flow

4.1. Cerințe funcționale

Cerințele funcționale ale aplicației SkyCast descriu în mod clar comportamentele așteptate ale sistemului în raport cu utilizatorul și sursele externe de date. Această aplicație meteo inteligentă are rolul de a colecta, procesa și interpreta date meteorologice provenite de la două servicii diferite - OpenMeteo și WeatherAPI - pentru a genera predicții personalizate, folosind algoritmi de regresie automatizați. În funcție de alegerea utilizatorului, predicțiile sunt generate pentru ziua curentă sau pentru ziua următoare, cu ajutorul unui model de învățare automată bazat pe unul dintre cei patru regresori disponibili: regresie liniară, random forest, boosted tree și decision tree.

Aceste cerințe stabilesc și modul în care rezultatele sunt afișate și interpretate de către utilizator. Predicțiile pentru temperatură, temperatură resimțită și probabilitatea de precipitații

sunt prezentate fie sub formă de listă textuală structurată, fie sub formă grafică. oferind o reprezentare vizuală a tendințelor orare. Funcționalitățile aplicației acoperă astfel întregul flux de la colectarea automată a datelor, agregarea acestora într-un model tabular, antrenarea algoritmului de predicție, până la evaluarea performanței modelului și prezentarea rezultatelor într-un format accesibil și intuitiv pentru utilizatorul final

- **Alegerea regresorului de antrenare a modelului:** Aplicația permite utilizatorului de a selecta regresorul cu care acesta dorește să antreneze modelul de date oferind astfel o flexibilitate mai mare la nivel de analiză, în funcție de complexitatea datelor, nevoia de interpretabilitate sau acuratețea dorită. Această funcționalitate este esențială pentru personalizarea modului de predicție, permițând testarea și compararea performanței fiecărui algoritm în raport cu parametrii meteorologici selectați.
- **Afișarea datelor sub diferite formate:** Atât datele preluate din cele două surse externe cât și cele prezise de către algoritmul de regresie selectat sunt afișate în format de listă pentru o vizualizare clară și aproximare mai bună a valorilor obținute. În plus, aplicația oferă și posibilitatea de a afișa, doar valorile prezise, sub formă grafică, facilitând identificarea tendințelor și variațiilor pe parcursul orelor selectate. Această dublă modalitate de prezentare, textuală și vizuală, contribuie la o experiență de utilizare mai accesibilă, adaptabilă preferințelor și stilului de interpretare al fiecărui utilizator.
- **Afișarea metricilor de evaluare:** Interfața grafică afișează de asemenea și rezultatele metricilor de evaluare în urma antrenării modelului pentru a indica performanța acestuia în raport cu datele reale. Printre metricile prezentate se numără eroarea absolută medie (MAE), eroarea pătratică medie (MSE), rădăcina pătrată a erorii pătratice medii (RMSE), și coeficientul de determinare (R^2). Aceste valori sunt afișate în partea de sus a interfeței, permițând utilizatorului să înțeleagă cât de bine a reușit modelul să prezică temperatura, senzația termică și probabilitatea de precipitații pentru ziua selectată. Evaluarea obiectivă a performanței este esențială pentru alegerea celui mai potrivit regresor în funcție de contextul analizat.
- **Posibilitatea de a alege ziua dorită:** Aplicația permite utilizatorului de a alege pentru ce zi își dorește să fie prezisă vremea: ziua curentă sau ziua următoare. Această funcționalitate oferă un grad sporit de control asupra orizontului temporal al predicției, adaptând analiza în funcție de nevoile utilizatorului, fie că acesta dorește să verifice acuratețea modelelor față de condițiile actuale, fie că intenționează să planifice activități pentru ziua următoare. Alegerea zilei influențează direct datele preluate de la sursele externe și instanța modelului care va fi antrenat, asigurând o predicție contextualizată și relevantă.
- **Configurarea parametrilor regresorului selectat:** Această funcționalitate permite utilizatorului posibilitatea de a ajusta parametrii specifici regresorului ales permițând un control mai fin asupra procesului de antrenare al modelului. Scopul acestor ajustări este de a îmbunătăți acuratețea predicțiilor generate. Eficiența modificărilor efectuate poate fi evaluată prin intermediul metricilor de performanță care indică în ce măsură setările alese contribuie la optimizarea modelului.

4.2. Cerințe non-funcționale

Cerințele non-funcționale ale aplicației SkyCast stabilesc modul în care sistemul trebuie să funcționeze din perspectiva performanței, fiabilității, securității și experienței utilizatorului. Aceste cerințe nu influențează direct funcționalitatea oferită, dar sunt esențiale pentru calitatea generală a aplicației și pentru satisfacția utilizatorului final. Având în vedere că aplicația lucrează cu modele de învățare automată, prelucrează volume de date meteorologice din surse externe și generează predicții dinamice, este esențial ca timpul de răspuns, claritatea interfeței și disponibilitatea sistemului să fie garantate în orice moment.

Aplicația trebuie să asigure performanță optimă, cu un timp scurt de preluare a datelor și generare a predicțiilor, chiar și în condiții de conexiune slabă sau instabilă. De asemenea, sistemul trebuie să fie scalabil, permițând extinderea viitoare a funcționalităților. Interfața grafică trebuie să fie intuitivă, coerentă și adaptabilă pentru a permite utilizarea fără efort din partea oricărui tip de utilizator, indiferent de nivelul tehnic. Aplicația trebuie să respecte standardele de securitate pentru gestionarea datelor utilizatorului și să mențină un nivel ridicat de disponibilitate fără erori critice care să afecteze fluxul de utilizare.

În plus, sistemul trebuie să fie ușor de întreținut și să permită actualizări periodice fără a afecta datele sau funcționalitatea deja implementată. Codul sursă ar trebui să fie bine documentat, modular și extensibil, iar componentele de prelucrare a datelor și antrenare a modelelor trebuie să fie testabile și reproductibile. Nu în ultimul rând, aplicația trebuie să genereze fișiere CSV într-un format standard compatibil cu alte instrumente externe de analiză, respectând principiile de portabilitate și interoperabilitate.

- **Securitatea:** Aplicația nu folosește date personale ale utilizatorului deloc pentru a își îndeplini scopul. Nu este nevoie nici de autentificare și nici de înregistrare ceea ce reduce semnificativ suprafața de atac din punct de vedere al securității cibernetice. Toate operațiunile se desfășoară local sau prin interacțiuni cu surse externe sigure, fără colectarea sau stocarea informațiilor sensibil. Această abordare minimalistă asigură confidențialitate deplină a utilizatorului și respectarea cerințelor legale privind protecția datelor, precum GDPR, fără a compromite funcționalitatea aplicației.
- **Utilizabilitatea:** SkyCast este concepută pentru a oferi o interfață prietenoasă, intuitivă și accesibilă, chiar și pentru utilizatorii fără experiență tehnică. Se pune accent pe claritatea vizuală a elementelor, pe navigarea logică între secțiuni și pe afișarea structurată a datelor, fie în format de listă, fie sub formă grafică. Utilizatorul are la dispoziție selecții simple, precum alegerea tipului de regresor sau a zilei pentru care dorește predicția, fără a fi necesare setări avansate sau intervenții compliate. Feedback-ul vizual este clar și imediat, fie în cazul unei erori, fie în momentul afișării predicțiilor sau metricilor de evaluare.
- **Scalabilitate:** Arhitectura aplicației este una modulară, care permite extinderea acesteia fără a afecta stabilitatea sistemului existent. Fie că se dorește integrarea de noi surse meteorologice, adăugarea altor algoritmi de învățare automată sau extinderea predicțiilor pe intervale mai lungi, aplicația poate fi adaptată ușor la noile cerințe. Separarea clară între componentele de colectare a datelor, procesare, antrenare și afișare oferă flexibilitate în dezvoltare și întreținere.
- **Portabilitate:** Din moment ce aplicația este dezvoltată în Swift și folosind tehnologii compatibile cu ecosistemul Apple, ceea ce asigură portabilitate între dispozitivele iOS și chiar între versiuni diferite de sistem de operare. De asemenea, fișierele generate de aplicație (CSV-urile cu datele din sursele

exterioare și rezultate de predicție) respectă formate standard, ușor de deschis și interpretat cu aplicații externe precum Excel, Numbers sau platforme de analiză statistică.

4.3. Prezentarea regresiiilor utilizate

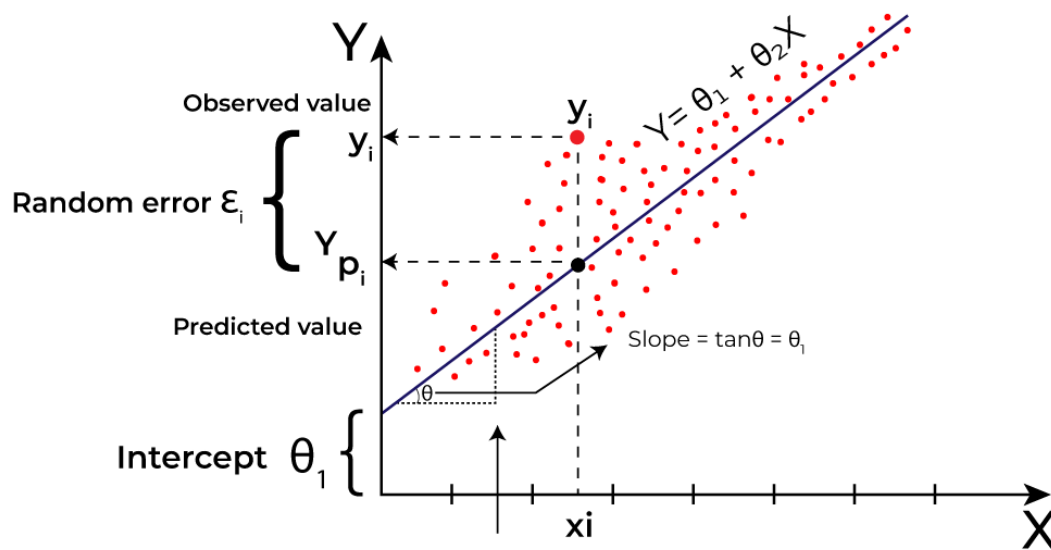
4.3.1. Regresie liniară

Regresia liniară este una dintre cele mai fundamentale și utilizare metode de modelare predictivă datorită simplității sale conceptuale și a eficienței în interpretarea relațiilor dintre variabile. În cadrul aplicației SkyCast, acest tip de regresor este folosit pentru a prezice temperatura, temperatura resimțită și probabilitatea de precipitații, pornind de la datele colectate orar din două surse externe. Alegerea regresiei liniare se bazează pe capacitatea sa de a estima valorile necunoscute printr-o combinație liniară a predictorilor, modelând astfel dependențele directe între valorile furnizate de OpenMeteo și WeatherAPI și valorile medii reale de la un anumit moment. Astfel, utilizatorul poate înțelege cum influențează fiecare sursă de date rezultatul final, ceea ce nu este întotdeauna posibil cu metodele mai complexe.

Potrivit studiului realizat de Kechen Qu [13], regresia liniară continuă să fie preferată în aplicațiile practice datorită clarității matematice, stabilității și ușurinței de interpretare, chiar dacă sunt disponibile modele mai sofisticate. Articolul evidențiază faptul că regresia liniară este eficientă în condițiile în care datele prezintă o relație aproape liniară, iar distribuția erorilor este relativ constantă. În aplicația SkyCast, această condiție este adesea îndeplinită datorită corelării ridicate dintre valorile prognozate de cele două surse și observațiile reale de la sol. Autorii subliniază că regresia liniară servește ca referință de bază în comparațiile cu alți algoritmi, fiind esențială pentru evaluarea îmbunătățirilor aduse de metodele non-liniare sau ensemble.

Implementarea regresiei liniare în aplicație este susținută de avantajele prezentate în articol: viteză de antrenare, consum redus de resurse și simplitate algoritmică, toate caracteristicile ideale pentru un sistem care rulează local pe dispozitive mobile. De asemenea, articolul propune utilizarea metricilor standard precum MAE, RMSE și R^2 , exact aceleași pe care aplicația SkyCast le afișează pentru ca utilizatorul să poată evalua performanța modelului selectat. În concluzie, regresia liniară nu doar că oferă o soluție practică pentru predicțiile meteo zilnice, dar se și aliniază perfect cu recomandările din literatura științifică actuală, reprezentând un echilibru între simplitate, transparență și utilitate operațională.

Formula standard a regresiei liniare simple este: $y = \beta_0 + \beta_1 x + \varepsilon$, y este variabila dependentă (ex: temperatura prezisă), x este variabila independentă (ex. temperatura de la OpenMeteo sau WeatherAPI), β_0 este interceptul (valoarea lui y când $x=0$), β_1 este coeficientul de regresie (arată cât se modifică y când x crește cu o unitate), iar ε este eroare reziduală (diferența dintre valoarea reală și cea prezisă).



Figură 4.2 - Regresia liniară [14]

4.3.2. Regresie „Random Forest”

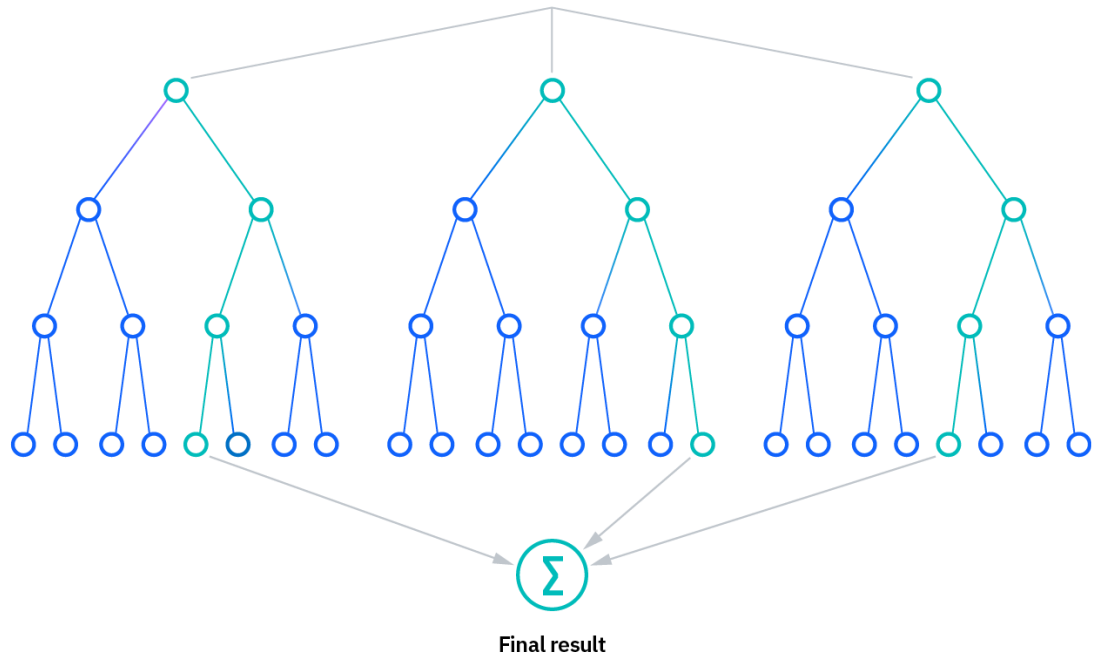
Random Forest este un algoritm de învățare automată de tip ensemble, care utilizează un ansamblu de arbori de decizie pentru a construi un model robust și rezistent la zgomotul din date. În loc să construiască un singur arbore mare, Random Forest generează sute sau mii de arbori mai mici, fiecare antrenat pe un subset aleator de date (bootstrap sampling) și cu un subset aleator de variabile la fiecare split. Predicția finală este obținută prin media aritmetică a rezultatelor oferite de toți arborii. Această abordare reduce semnificativ varianța modelului și îl face mai puțin predispus la overfitting, în special în comparație cu un singur arbore de decizie. Datorită acestei robusteți, Random Forest este adesea utilizat în scenarii cu relații non-liniare, seturi de date eterogene sau lipsă parțială a valorilor.

Formula matematică este $\hat{y} = \frac{1}{N} \sum_{i=1}^N T_i(x)$, unde \hat{y} este valoarea prezisă, $T_i(x)$ este predicția arborelui de decizie i pentru vectorul de intrare x , iar N este numărul total de arbori din pădure.

În cadrul aplicației SkyCast, Random Forest este unul dintre cei patru regresori disponibili pentru predicția temperaturii, temperaturii resimțite și probabilității de precipitații. Spre deosebire de regresia liniară, care presupune o relație simplă și continuă între predictor și valoarea țintă, Random Forest poate modela variații bruște și interacțiuni complexe dintre variabilele provenite din sursele OpenMeteo și WeatherAPI. Astfel, atunci când datele colectate prezintă discrepanțe sau zgomot (ex. valori anormale sau lipsuri), Random Forest tinde să fie mai rezistent și oferă rezultate mai precise, în special pe termen scurt.

Un avantaj major al Random Forest este că nu presupune nicio relație liniară între variabile și poate modela dependențe complexe fără transformări suplimentare. De asemenea, poate estima importanța fiecărei variabile în model, oferind astfel informații utile despre influența fiecărei surse de date. În schimb, algoritmul este mai dificil de interpretat decât un model liniar, ceea ce îl face mai greu de explicat utilizatorilor obișnuiți. De asemenea, este mai costisitor computațional, necesitând mai mult timp și memorie pentru antrenare și inferență, aspect care poate influența performanța aplicației pe dispozitive mobile, dacă nu este optimizat corespunzător.

Studiul „Weather prediction using random forest machine learning model” [15] confirmă eficiența acestui algoritm în predicția parametrilor meteorologici, cum ar fi radiația solară, temperatura și viteza vântului. Comparativ cu alte metode (SVM, rețele neuronale), Random Forest a înregistrat un scor ridicat R^2 și o eroare pătratică medie scăzută (MSE), demonstrând superioritatea sa în scenarii cu date zgomotoase sau complexe.



Figură 4.3 - Random Forest [16]

4.3.3. Regresie „Boosting Tree”

Regresia Boosted Tree este un algoritm de învățare automată de tip ensemble, unde arborii de decizie sunt construiți secvențial, fiecare încercând să corecteze erorile modelului anterior. În contrast cu Random Forest (care folosește bootstrap pentru training), Boosting adaugă arbori noi focalizați pe predicțiile anterioare greșite, astfel fiecare arbore reduce bias-ul modelului. Acest proces iterativ face ca Boosted Tree să capteze modelele complexe și neliniare, îmbunătățind precizia predicțiilor în scenarii meteo variabile.

Formula matematică este $F_m(x) = F_{m-1}(x) + v \cdot h_m(x)$, unde $F_m(x)$ este predicția modelului după m arbori, v este rata de învățare, iar $h_m(x)$ este arborele adăugat la pasul m , ajustat pentru a corecta eroarea modelului anterior.

În aplicația SkyCast, Boosted Tree reprezintă o opțiune avansată de regresie. Se antrenează folosind setul de date care combină parametri de la OpenMeteo și WeatherAPI, ajustând arborii noi pentru fiecare eroare detectată. Această metodă este utilă când temperatura sau probabilitatea de precipitații nu urmează un tren simplu și există interacțiuni complexe între predicatori. Modelul poate îmbunătăți rezultatele obținute cu regresia liniară sau Random Forest prin reducerea sistematică a erorilor succesive.

Câteva avantaje ale acestei metode sunt precizia ridicată și adaptabilitatea la relații complexe și capacitatea de a captura nelinearități subtile. Pe de altă parte există și câteva dezavantaje precum necesitatea de tuning de hiperparametrii (learning rate, număr de arbori) datorită gradului ridicat de risc al overfitting-ului și faptul că este mult mai costisitor din punct de vedere computațional (poate necesita preprocesare și validare riguroasă).

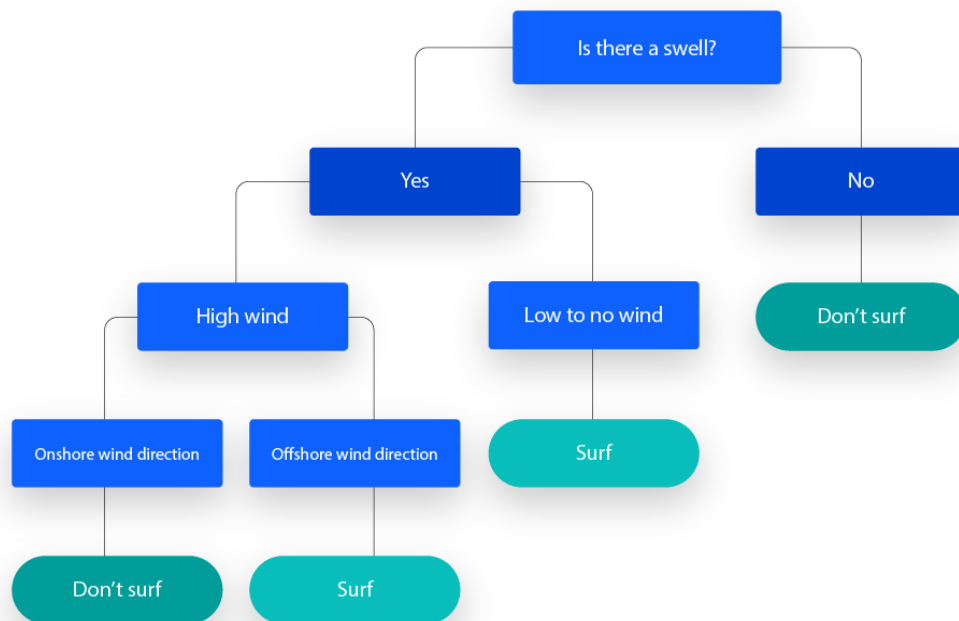
4.3.4. Regresie „Decision Tree”

Regresia de tip arbori de decizie construiește un arbore de decizie în care în fiecare nod rezolvă întrebări despre valorile predictorilor (ex. „temperatura > 14 grade? ”), până când frunzele indică o valoare numerică prezisă. Fiind ușor de interpretat, acest model este util în aplicații unde transparența contează. Cu toate acestea, arborii adânci pot suferi de overfitting memorând zgomotul din date în loc să învețe pattern-uri semnificative.

Nu există o formulă prestabilită pentru acest tip de regresie întrucât regulile sunt de tip if-else; la fiecare nod se alege împărțirea care minimizează MSE în subseturile obținute. Fiecare frunză conține media țințelor din acel subset.

În aplicația realizată pentru această lucrare, regresia cu arbori de decizie este o opțiune intermediară între modelul liniar și cele de tip ensemble. Poate fi folosit pentru a explica modul în care un anumit predictor (ex. temperatura de la WeatherAPI) determină praguri de decizie. Este rapid de antrenat, dar poate genera predicții mai volatile, în special când datele meteo sunt variabile. Pentru a evita overfitting-ul, implementarea ar trebui completată cu limită de adâncime sau număr minim de exemple pe frunză.

Câteva avantaje sunt interpretabilitatea foarte bună (structura poate fi vizualizată și înțeleasă de utilizatori) și este rapid de construit și folosește resurse reduse. Cu toate acestea, acest tip de regresie conține și câteva limitări și anume predicții discontinue și sensibile la variații mici în date și necesitatea tehnicilor de regular (pruning, maxDepth) pentru a preveni overfitting.



Figură 4.4 - Regresie cu arbori de decizie [17]

4.4. Metricile de evaluare utilizate

4.4.1. Eroarea medie absolută (MAE)

Eroarea medie absolută indică în medie cu cât se abat valorile prezise de cele reale, fără a lua în calcul direcția diferenței (pozitivă sau negativă). Este o metrică intuitivă, ușor de interpretat și oferă o estimare clară a greșelii comise de model. De exemplu, o valoarea MAE de 2,5 ar însemna că, în medie, modelul estimează temperatura cu o eroare de $\pm 2,5^{\circ}\text{C}$.

Formula acestia este: $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$.

4.4.2. Eroarea medie pătratică (MSE)

Comparativ cu eroarea medie absolută, eroarea medie pătratică are rolul de a penaliza mai sever abaterile mari dintre predicții și realitate, prin ridicarea la pătrat a diferenței. Această abordare face ca erorile mari să aibă un impact disproporționat în calculul final, ceea ce poate fi util în detecția regresorilor care uneori generează deviații semnificative. MSE este totuși mai greu de interpretat direct având o unitate diferită față de variabila prezisă.

Formula matematică a acesteia este: $MAE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$.

4.4.3. Root Mean Squared Error

Aplicația folosește și RMSE care reprezintă rădăcina pătratică a MSE. Această metrică exprimă eroarea într-o unitate comparabilă cu valorile reale (ex. grade Celsius pentru temperatură), fiind mai ușor de înțeles pentru utilizatorii care vor să evalueze direct cât de departe sunt predicțiile de valorile reale. Astfel, RMSE combină sensibilitatea la erori mari cu claritatea interpretării numerice.

Formula matematică este: $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$

4.4.4. Coeficientul de determinare

Acesta reflectă cât de bine explică modelul variațiile din datele reale. Această valoare se situează, în mod ideal, între 0 și 1, unde 1 înseamnă că modelul a prezis perfect toate valorile, iar 0 că nu a fost mai bun decât o medie constantă. În unele cazuri, R^2 poate fi chiar negativ, semnalând că modelul este mai slab decât o estimare trivială. În SkyCast, un R^2 ridicat pentru temperatură, dar scăzut pentru precipitații, poate indica că datele meteo sunt mai previzibile într-un caz decât altul.

Formula matematică este: $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$, unde $SS_{res} = \sum (y_i - \hat{y}_i)^2$ (suma pătratelor erorilor (reziduurilor)) și $SS_{tot} = \sum (y_i - \bar{y})^2$ (suma pătratelor deviațiilor față de media valorilor reale)

4.5. Structura proiectului

4.5.1. Structura fișierelor

Proiectul a fost organizat pe module și directoare denumite sugestiv pentru a permite o navigare cât mai intuitivă între fișiere. Pentru gestionarea cererilor de rețea și obținerea modelelor de date oferite de OpenMeteo și WeatherAPI, a fost utilizat pachetul extern „Alamofire”. În plus, a fost creat un modul propriu, denumit „Constants”, care are rolul de a defini baza URL-urilor folosite în interogările de rețea, precum și coordonatele geografice (latitudine și longitudine) pentru locația vizată. Acest director include și o extensie dedicată tipului primitiv „Calendar”, ce oferă funcționalități utile precum obținerea datei curente sau a zilei următoare.

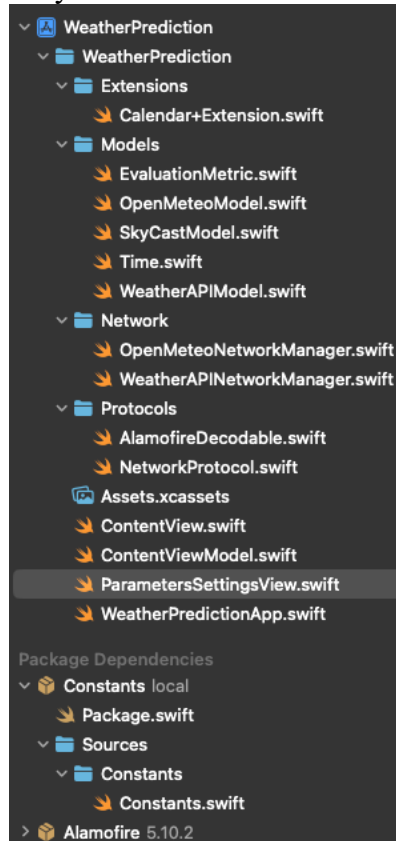
Directorul „Models” include toate structurile de date utilizate în aplicație, acoperind atât modelele specifice surselor externe (WeatherAPI și OpenMeteo), cât și modelul antrenat final, structurat conform clasei „SkyCastModel”, denumire care reflectă clar forma tabelară a datelor (CSV) folosite pentru antrenare și predicție.

În cadrul directorului „Network” se regăsesc cele două clase principale responsabile pentru preluarea și decodificarea informațiilor meteorologice provenite de la cele două API-

uri externe. Aceste date stau la baza procesului de construire a datasetului combinat și a antrenării modelului de predicție.

Directorul „Protocols” conține două protocoale Swift (echivalentul interfețelor din Java): primul, „AlamofireDecodable”, este folosit pentru a standardiza decodificarea răspunsurilor API, iar al doilea asigură o decuplare clară a logicii de rețea fiind implementat de clasele din modulul „Network”.

În cele din urmă, fișierul „ContentView” conține interfața grafică a aplicației prin care utilizatorul interacționează cu sistemul, iar „ContentViewModel” gestionează logica principală: colectarea datelor, procesarea acestora și antrenarea efectivă a modelului de predicție utilizat în cadrul aplicației SkyCast.



Figură 4.5 - Structura fișierelor

4.5.2. Structura interfeței grafice

Interfața grafică a aplicației SkyCast a fost proiectată pentru a oferi utilizatorului o experiență intuitivă, clar structurată și interactivă în procesul de analiză a predicțiilor meteorologice generate prin intermediul diverselor modele de regresie.

În partea superioară stângă a ecranului se regăsesc două meniuri principale: „Actions” și „Configuration menu”. Meniul „Actions” permite declanșarea procesului de predicție prin opțiunea „Predict with selected regressor”, care antrenează modelul selectat și generează predicția aferentă. De asemenea, include și funcționalitatea „Re-fetch data and make prediction”, prin care utilizatorul poate actualiza datele meteo și relua procesul de predicție în baza celor mai recente informații disponibile.

Meniul „Configuration menu” oferă posibilitatea selectării tipului de regresor utilizat în analiză, cum ar fi spre exemplu „Boosting Tree Regression”, precum și alegerea zilei pentru care se dorește efectuarea predicției, fie pentru ziua curentă, fie pentru ziua următoare. Această componentă sporește flexibilitatea aplicației, permițând compararea rezultatelor obținute în funcție de algoritmul ales și intervalul temporal vizat. În partea dreaptă a aceleiași

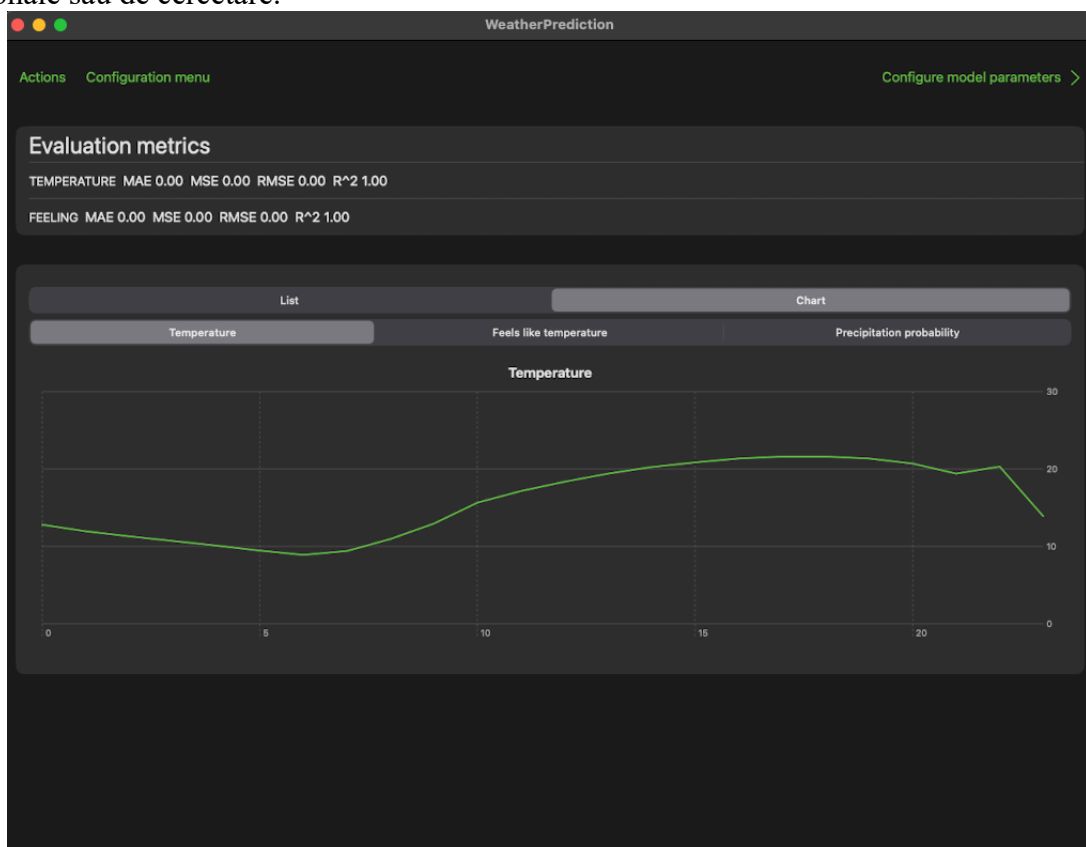
secțiuni superioare se regăsește butonul „Configure model parameters”, care redirecționează utilizatorul către o pagină dedicată configurării detaliate a modelului selectat. Aici pot fi ajustați parametrii algoritmului în vederea îmbunătățirii preciziei predicției.

Secțiunea intitulată „Evaluation metrics” afișează valorile celor patru indicatori de performanță (MAE, MSE, RMSE și R^2) pentru fiecare dintre cei trei parametri meteorologici: temperatură, temperatură resimțită și probabilitate de precipitații. Este important de menționat că pentru precipitații este inclusă o mențiune suplimentară („non-zero only”) ceea ce indică faptul că evaluarea a fost efectuată doar pe intervalele orare în care probabilitatea nu a fost zero, sporind relevanța metricilor.

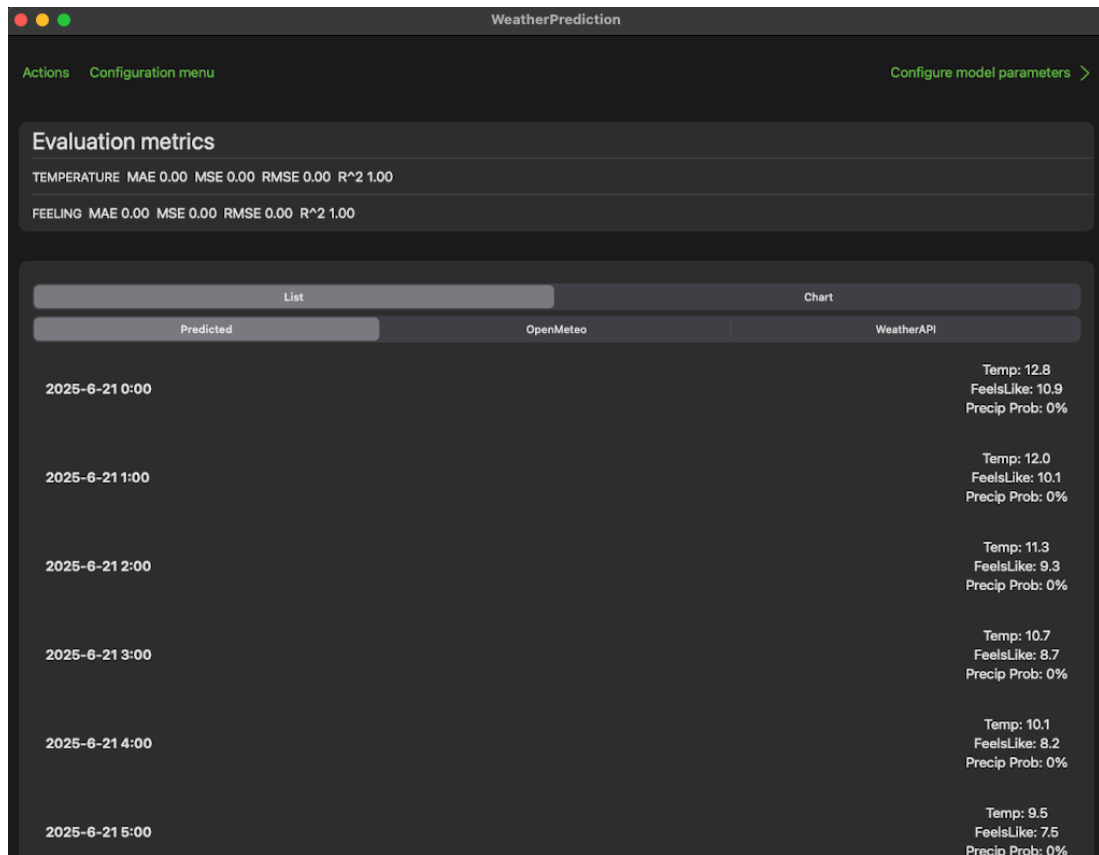
În zona inferioară a ecranului, interfața este împărțită în două taburi: „List” și „Chart”. În primul tab, utilizatorul poate vedea, într-o formă tabelară, valorile predicției pentru fiecare oră din zi, alături de valorile extrase din cele două surse externe, OpenMeteo și WeatherAPI. Informațiile sunt afișate ordonat, cu timestamp vizibil și valori explicite pentru temperatură, senzație termică și probabilitate de precipitații, astfel încât utilizatorul să poată compara cu ușurință cele trei seturi de date.

În cel de al doilea tab, „Chart”, aplicația generează un grafic orar interactiv pentru fiecare dintre cei trei parametri meteorologici, în funcție de selecția utilizatorului. De exemplu, în prima imagine este vizualizată temperatura pe parcursul zilei, sub formă de linie continuă, ceea ce permite detectarea tendințelor sau variațiilor pe intervale orare. Axele sunt clare și etichetate, iar prezentarea este minimalistă, în ton cu estetica generală a aplicației.

Per total, interfața SkyCast reușește să echilibreze complexitatea tehnică a unui sistem de predicție cu accesibilitatea pentru utilizatorul final. Prin organizarea logică a componentelor, afișarea clară a metricilor și posibilitatea de comparare vizuală și tabelară a datelor, aplicația oferă un instrument valoros atât pentru uz personal, cât și pentru scopuri educaționale sau de cercetare.



Figură 4.6 - Interfața grafică în modul grafic



Figură 4.7 - Interfața grafică în modul listă

4.6. Structura model de date

Deși structura datelor furnizate de cele două surse externe, OpenMeteo și WeatherAPI, diferă semnificativ în ceea ce privește organizarea și denumirea câmpurilor, s-a urmărit în mod constant optimizarea reutilizării codului și a datelor în procesul de construcție a modelului final. Prin abstractizarea logicii de extragere și uniformizarea formatelor relevante, dezvoltatorul proiectului a reușit să evite redundanțele și să simplifice procesul de integrare.

S-a pus accent pe crearea unor structuri cât mai comune și funcții generice care să prelucreze coerent datele provenite din ambele surse, facilitând astfel combinarea eficientă a informațiilor într-un singur set de antrenament. Această abordare a permis nu doar crearea rapidă a modelului final utilizat pentru predicție, ci și o scalabilitate mai mare a aplicației în eventualitatea adăugării de noi surse externe în viitor.

Un aspect important este modul de decodare a datelor. Codul folosește metode sigure precum `decodeIfPresent` pentru a evita erorile în cazul în care anumite câmpuri lipsesc.

4.6.1. Structura modelului de date OpenMeteo

Modelul de date furnizat de OpenMeteo este respectat prin structura definită în aplicație sub denumirea `OpenMeteoModel`. Această structură conține proprietățile latitudine și longitudine, precum și o componentă de tip `Hourly`, care grupează toate valorile meteo relevante sub formă de vectori paraleli. Mai precis, structura `Hourly` include patru proprietăți esențiale: un vector de date (`time`) și trei vectori de valori numerice: `temp` (temperatura), `feelLikeTemp` (temperatura resimțită) și `precipProb` (probabilitatea de precipitații). Fiecare poziție din acești vectori corespunde unei ore din zi, ceea ce înseamnă că datele sunt aliniate pozițional, adică valorile de la indexul `i` din toți vectorii se referă la aceeași oră.

Aceasta abordare presupune o coerență strictă a pozițiilor între vectori. Spre exemplu, dacă `time[3]` indică ora 12:00, atunci `temp[3]`, `feelLikeTemp[3]` și `precipProb[3]` reprezintă valorile meteorologice aferente acelei ore. Prin urmare, în momentul construirii modelului final de antrenament, aplicația parcurge toate aceste valori simultan, pe baza indexului comun, pentru a asambla un rând complet de date pe oră.

În concluzie, modelul `OpenMeteoModel` a fost gândit astfel încât să reflecte fidel structura JSON a răspunsului API-ului `OpenMeteo`, asigurând o mapare directă între datele sursă și cele procesate pentru predicție.

```
struct Hourly: Decodable, DataDecoder, AlamofireDecodable {
    let time: [Time]
    let temp: [Double]
    let feelLikeTemp: [Double]
    let precipProb: [Int]

    enum CodingKeys: String, CodingKey {
        case time
        case temp = "temperature_2m"
        case feelLikeTemp = "apparent_temperature"
        case precipProb = "precipitation_probability"
    }

    init(from decoder: Decoder) throws {
        let container = try decoder.container(keyedBy: CodingKeys.self)

        let dateStringArray = try container.decodeIfPresent([String].self, forKey: .time) ?? []

        self.time = dateStringArray.compactMap { Time(from: $0, withDateFormat: Constants.OpenMeteo.dateFormat) }
        self.temp = try container.decodeIfPresent([Double].self, forKey: .temp) ?? []
        self.feelLikeTemp = try container.decodeIfPresent([Double].self, forKey: .feelLikeTemp) ?? []
        self.precipProb = try container.decodeIfPresent([Int].self, forKey: .precipProb) ?? []
    }
}

struct OpenMeteoModel: Decodable, DataDecoder, AlamofireDecodable {
    let latitude: Double
    let longitude: Double
    let hourly: Hourly

    enum CodingKeys: String, CodingKey {
        case latitude
        case longitude
        case hourly
    }

    init(from decoder: Decoder) throws {
        let container = try decoder.container(keyedBy: CodingKeys.self)
        self.latitude = try container.decodeIfPresent(Double.self, forKey: .latitude) ?? 0.0
        self.longitude = try container.decodeIfPresent(Double.self, forKey: .longitude) ?? 0.0
        self.hourly = try container.decodeIfPresent(Hourly.self, forKey: .hourly) ?? Hourly(from: decoder)
    }
}
```

Figură 4.8 - Structura modelului de date `OpenMeteoModel`

4.6.2. Structura modelului de date `WeatherAPI`

Datele meteorologice obținute prin API-ul `WeatherAPI` sunt structurate într-un mod mai ierarhizat, față de modelul liniar al `OpenMeteo`. Structura principală, denumită `WeatherAPIModel`, conține două componente esențiale: `location`, care stochează datele geografice (latitudine și longitudine), și `forecast` care include predicțiile meteo grupate pe zile (`forecastday`). Fiecare obiect `ForecastDay` conține un vector de tip `hour`, unde fiecare element corespunde unei ore și este modelat prin structura `Hour`.

Spre deosebire de `OpenMeteo`, unde toți parametrii sunt grupați în vectori paraleli, aici fiecare oră este reprezentată printr-un obiect individual care conține toate valorile asociate momentului respectiv: ora (`time`), temperatura (`temp_c`), temperatura resimțită (`feelslike_c`) și probabilitatea de precipitații (`chance_of_rain`). Aceste câmpuri sunt mapate prin intermediul `CodingKeys` pentru a permite o conversie corectă din formatul JSON în structuri Swift.

De asemenea, și în decodificarea acestui model se folosește `decodeIfPresent` care oferă valori implicite în caz de absență al acestora. Spre exemplu, dacă valoarea de temperatură lipsește, aceasta este completată cu 0.0.

Această structura obiectuală în care fiecare oră are propriul set de valori meteo, oferă o accesare clară și segmentată a datelor, ceea ce este extrem de util în procesul de aliniere cu vectorii `OpenMeteo` pentru antrenarea modelului final. În momentul integrării, aplicația parcurge toate obiectele `hour` din fiecare `ForecastDay` și le corelează cu elementele corespondente din modelul `OpenMeteo`, folosind ca reper comun câmpul `time`.

```
struct Hour: Decodable, DataDecoder, AlamofireDecodable {
    let time: Time
    let temp: Double
    let feelLikeTemp: Double
    let precipProb: Int

    enum CodingKeys: String, CodingKey {
        case time
        case temp = "temp_c"
        case feelLikeTemp = "feelslike_c"
        case precipProb = "chance_of_rain"
    }

    init(from decoder: Decoder) throws {
        let container = try decoder.container(keyedBy: CodingKeys.self)

        let dateStringArray = try container.decodeIfPresent(String.self, forKey: .time) ?? ""

        self.time = Time(from: dateStringArray, withDateFormat: Constants.WeatherAPI.dateFormat) ?? Time(from: "999-01-01 00:00", withDateFormat: Constants.WeatherAPI.dateFormat)!
        self.temp = try container.decodeIfPresent(Double.self, forKey: .temp) ?? 0
        self.feelLikeTemp = try container.decodeIfPresent(Double.self, forKey: .feelLikeTemp) ?? 0
        self.precipProb = try container.decodeIfPresent(Int.self, forKey: .precipProb) ?? 0
    }
}

struct ForecastDay: Decodable, DataDecoder, AlamofireDecodable {
    let hour: [Hour]
}

struct Forecast: Decodable, DataDecoder, AlamofireDecodable {
    var forecastday: [ForecastDay] = []
}

struct Location: Decodable, DataDecoder, AlamofireDecodable {
    var lat: Double = 0
    var lon: Double = 0
}

struct WeatherAPIModel: Decodable, DataDecoder, AlamofireDecodable {
    var location: Location = Location()
    var forecast: Forecast = Forecast()

    enum CodingKeys: String, CodingKey {
        case location
        case forecast
    }
}
```

Figură 4.9 - Structura modelului de date `WeatherAPIModel`

4.6.3. Structura modelului de date `SkyCast`

Modelul de date final utilizat în aplicația `SkyCast` pentru antrenare și predicție este definit prin structura `SkyCastModel`. Acesta este un tip de obiect care respectă protocoalele `Identifiable` și `Hashable`, ceea ce îl face compatibil atât cu afișarea în liste dinamice `SwiftUI`, cât și cu operații precum gruparea sau compararea între instanțe. Fiecare instanță a acestui model are un identificator unit, `id`.

Structura `SkyCastModel` este concepută astfel încât să unifice datele provenite din cele două surse externe, `OpenMeteo` și `WeatherAPI`, și să le combine într-un singur obiect standardizat, utilizat pentru antrenarea modelelor de regresie și pentru afișarea rezultatelor prezise. Câmpurile `latitude`, `longitude` și `time` descriu coordonatele spațiale și temporale.

Modelul este împărțit logic în trei secțiuni de date:

- Prima secțiune conține datele provenite de la `OpenMeteo`: temperatura (`omTemp`), temperatura resimțită (`omFeelLike`) și probabilitatea de precipitații (`omPrecipProb`).
- A doua secțiune înregistrează aceiași parametri, dar din sursa `WeatherAPI`: `wTemp`, `wFeelLike` și `wPrecipProb`.
- A treia secțiune conține valorile prezise de modelul antrenat: `pTemp`, `pFeelLike` și `pPrecipProb`.

Această structură permite nu doar agregarea datelor de intrare, ci și compararea directă între surse și rezultate, ceea ce este esențial în evaluarea performanței algoritmilor de regresie. Fiecare instanță `SkyCastModel` corespunde unei ore din zi și este construită pe baza

alinierii temporale a valorilor de la cele două surse, urmând ca modelul să prezică parametrii meteorologici pentru acel moment specific.

În plus, fiind proiectat pentru export sub formă de fișier CSV, acest model este compatibil cu antrenarea automată în cadrul Apple CreateML și poate fi analizat ulterior cu instrumente externe ceea ce adaugă un plus de versatilitate aplicației.

```
struct SkyCastModel: Identifiable, Hashable {
    var id: String {
        UUID().uuidString
    }

    var latitude: Double
    var longitude: Double
    var time: Time?

    // OpenWeather data
    var omTemp: Double = 0
    var omFeelLike: Double = 0
    var omPrecipProb: Int = 0

    // WeatherAPI data
    var wTemp: Double = 0
    var wFeelLike: Double = 0
    var wPrecipProb: Int = 0

    // Predicted data
    var pTemp: Double = 0
    var pFeelLike: Double = 0
    var pPrecipProb: Double = 0

    func hash(into hasher: inout Hasher) {
        hasher.combine(id)
    }
}
```

Figură 4.10 - Structura modelului de date SkyCastModel

4.7. Antrenarea modelului

4.7.1. Crearea și antrenarea modelului folosind Create ML

Compania Apple a urmărit integrarea tuturor sistemelor sale de operare într-un ecosistem unitar, dorind să asigure o comunicare fluentă între toate dispozitivele pe care le comercializează, de la iPhone și iPad până la MacBook și Apple Watch. Pentru a susține această dorință de interoperabilitate, Apple oferă dezvoltatorilor o serie de instrumente specializate care facilitează crearea de aplicații optimizate pentru toate aceste platforme. Printre aceste instrumente se numără și Create ML, un mediu dedicat dezvoltării de modele de inteligență artificială, integrat direct în cadrul IDE-ului Xcode.

Create ML oferă suport pentru antrenarea unei game variate de modele AI precum: detecție de obiecte, clasificare de imagini, clasificare de sunete, învățare pe seturi tabulare, și ceea ce este relevant în aplicația SkyCast, regresie tabulară. Un aspect important al Create ML este că procesul de antrenare are loc direct pe dispozitiv, fiind compatibil doar cu macOS (deci este necesar un MacBook, MacMini sau MacPro pentru rulare). Acest detaliu are implicații importante: dacă se dorește actualizarea zilnică a modelului cu date noi, este necesar ca modelul să fie reantrenat zilnic pe unui dintre aceste dispozitive și apoi integrat manual în aplicația iOS.

Procesul începe prin generarea programatică a unui fișier CSV, care conține datele meteorologice utilizate la antrenare: temperatura, temperatura resimțită și probabilitatea de precipitații. Acest fișier este importat în CreateML, unde poate fi însoțit, opțional și de fișiere pentru validare și testare. Totuși, în practică, pentru că validarea ar necesita colectarea manuală zilnică a datelor reale (din Apple Weather și Google Weather), s-a ales varianta de validare automată oferită de platformă. În acest mod, Create ML împarte automat datele de

antrenare într-un subset pentru validare. În ceea ce privește testarea, nu există în această implementare date dedicate și nicio opțiune automatizată pentru generarea lor.

Următorul pas presupune alegerea coloanei țintă (adică variabila de prezis Target) și a coloanelor care servesc drept predictorii (Features). După definirea acestor seturi, se trece în secțiunea „Parameters” unde se selectează algoritmul de regresie dorit. Create ML oferă cinci opțiuni de regresori, fiecare cu un set propriu de hiperparametri configurabili:

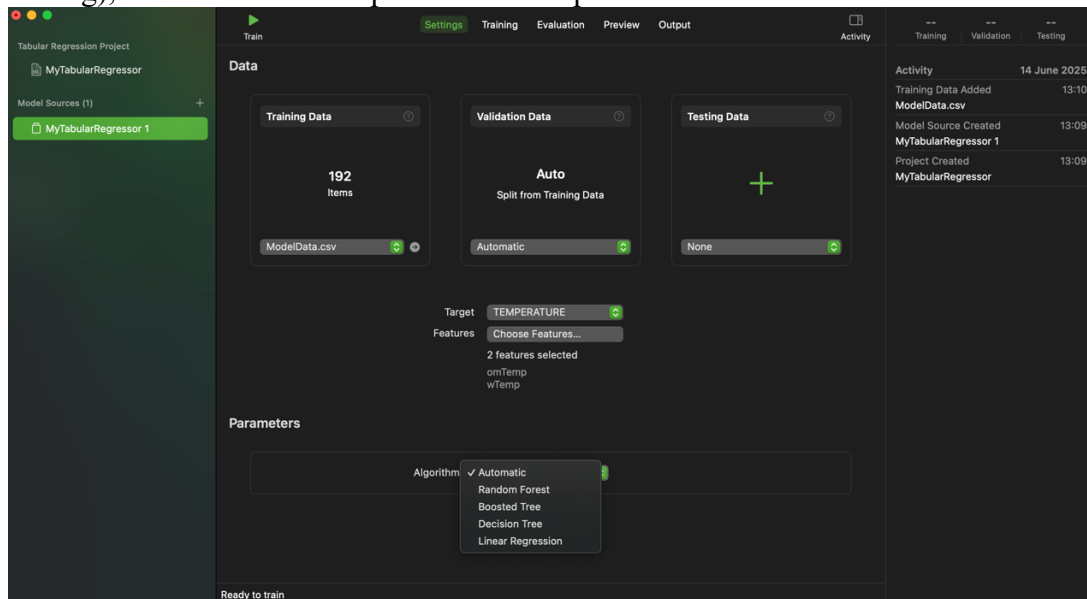
- Automatic: sistemul alege automat algoritmul optim împreună cu valorile parametrilor aferenți pe baza datelor disponibile.

- Random Forest: permite ajustarea unor parametri precum: numărul maxim de iterații (10), adâncimea maximă a arborilor (6), reducerea minimă a pierderilor (0), greutatea minimă a unui nod copil (0.1), raportul de subeșantionare pe rânduri (0.8) și pe coloane (0.8).

- Boosted Tree: include toți parametrii disponibili în Random Forest cu adăugarea unui parametru suplimentar și anume mărimea pasului (step size) implicit setată la 0.3, care controlează cât de mult contribuie fiecare arbore nou la modelul final.

- Decision Tree: un algoritm simplificat care permite configurarea a trei parametri esențiali: adâncimea maximă (6), reducerea pierderilor (0) și greutatea minimă a nodului copil (0.1).

- Regresie Liniară: acest algoritm include hiperparametri specifici: numărul maxim de iterații (10), penalizările L1 (0) și L2 (0.01), folosite pentru regularizare, mărimea pasului (1) și pragul de convergență (0.01). În plus, există o opțiune de activare a scalării caracteristicilor (feature scaling), utilă mai ales când predictorii sunt pe scări de valori diferite.



Figură 4.11 - Pregătirea modelului folosind Create ML

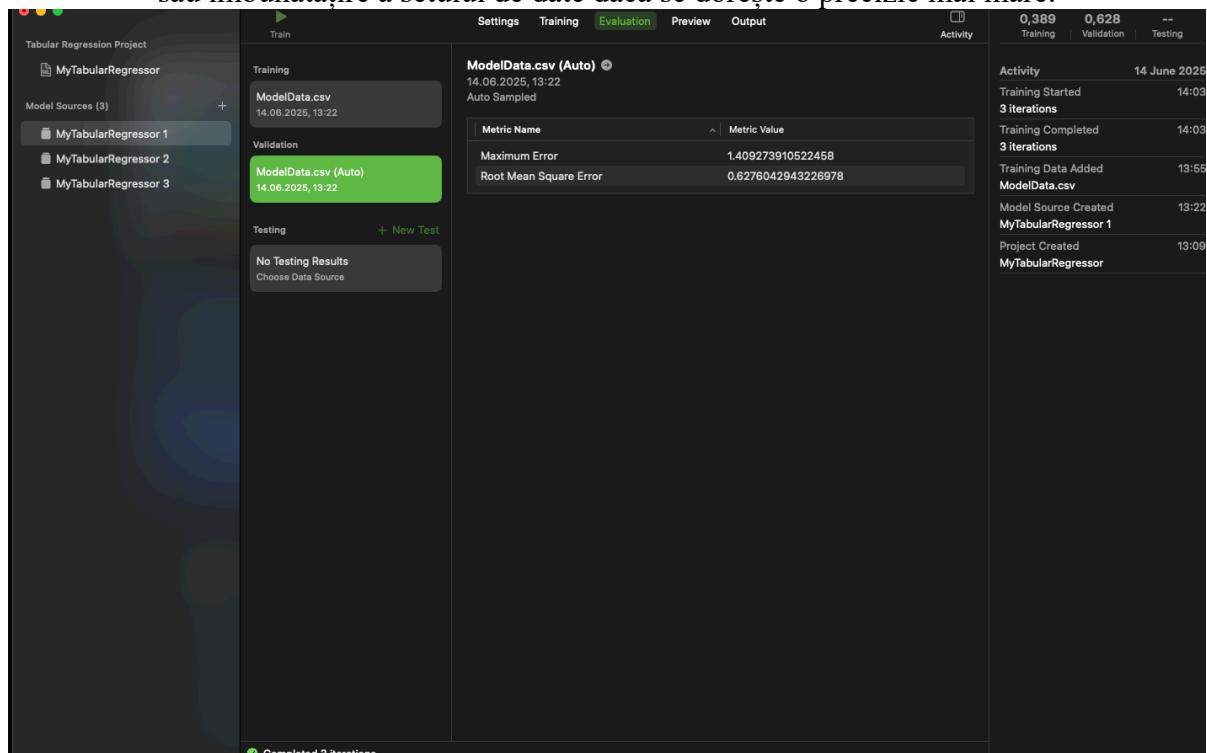
În figura 4.10 este ilustrat modul în care datele meteorologice, prelucrate și salvate anterior sub formă de fișier CSV, sunt încărcate în interfața grafică a instrumentului Create ML. Fișierul este introdus în secțiunea „Training Data”, care reprezintă sursa principală de informații pentru antrenarea modelului. După import, coloana care trebuie prezisă, în acest caz Temperature, este selectată ca „Target”, adică variabila dependentă. Predictorii (sau „Features”) aleși pentru antrenare sunt omTemp care reprezintă temperatura oferită de OpenMeteo și wTemp care este temperatura furnizată de WeatherAPI. Aceste două coloane oferă datele de intrare pe baza cărora modelul va învăța să prezică temperatura reală.

Inițial, algoritmul este configurat pe opțiunea „Automatic”, ceea ce înseamnă că Create ML va selecta automat cel mai potrivit regresor și valorile implicite ale hiperparametrilor în funcție de structura și volumul setului de date. Cu toate acestea,

utilizatorul are libertatea de a schimba manual algoritmul, putând alege între regresori precum Random Forest, Boosted Tree, Decision Tree sau regresie liniară. După finalizarea configurării câmpurilor esențiale și alegerea algoritmului dorit, în partea inferioară a ferestrei apare mesajul „Ready to train”, semnalând că totul este pregătit pentru începerea procesului de antrenare. Utilizatorul poate porni acest proces prin apăsarea butonului „Run”, moment în care este redirectionat automat către pagina de antrenare.

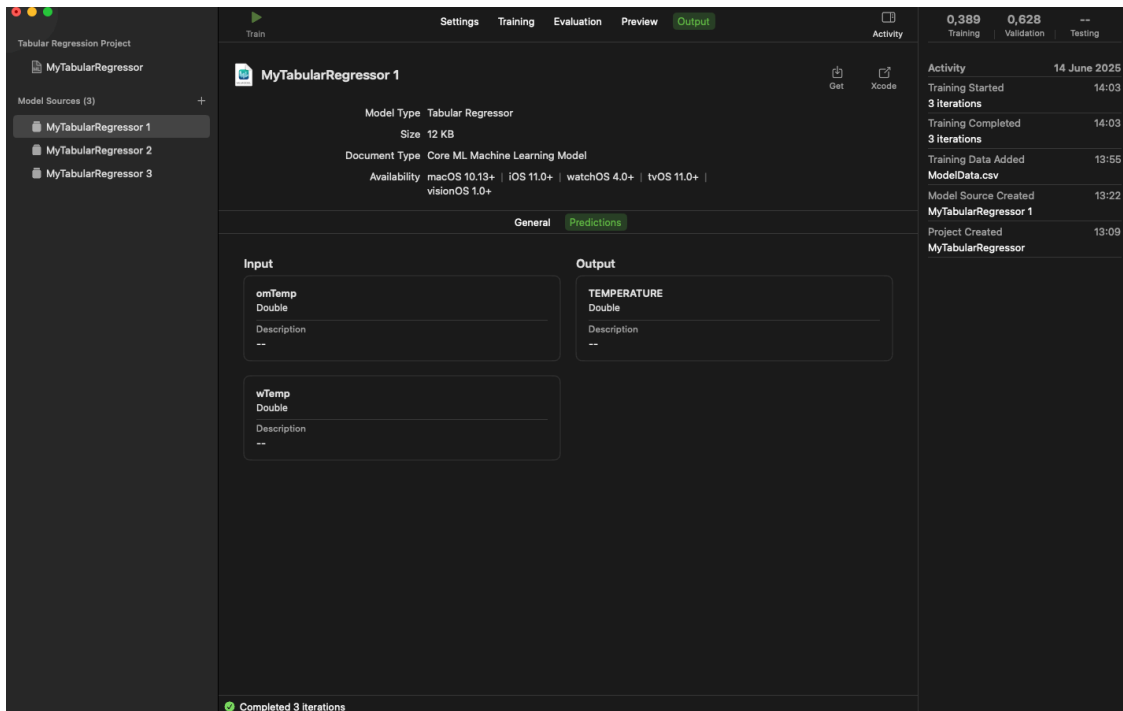
Pe durata antrenării, Create ML trece într-un mod de execuție dedicat, în care monitorizează progresul și ajustează modelul intern pe baza datelor furnizate. La final, utilizatorul este direcționat către pagina „Evaluation” care conține o analiză sumară a performanței modelului. În această secțiune, sunt prezentate două metrici de validare standard: eroare maximă (Max Error) și rădăcina pătrată a erorii medii pătratice (RMSE). Aceste valori oferă o imagine imediată asupra calității modelului și a capacității sale de generalizare.

În figura următoare, putem observa un exemplu în care algoritmul Random Forest a fost utilizat pentru antrenare. Modelul obținut are o eroare maximă de 1.4 ceea ce înseamnă că pentru cel puțin o observație, predicția modelului s-a abătut de la valoarea reală cu 1.4 unități, un indiciu privind cea mai mare greșeală comisă în setul de validare. În plus, RMSE-ul raportat este de 0.6, ceea ce indică faptul că în medie, modelul greșește cu 0.6 unități. Această valoare, aflată într-un interval rezonabil pentru predicții meteo, sugerează că modelul oferă estimări destul de precise, dar totodată poate semnala nevoia de ajustare a parametrilor sau îmbunătățire a setului de date dacă se dorește o precizie mai mare.



Figură 4.12 - Evaluarea modelului folosind Create ML

În ultima fereastră, Output, se pot observa mai multe detalii în legătură cu modelul obținut precum dimensiunea acestuia, tipul, care au fost datele de intrare și care au fost datele de ieșire, cât la sută din datele de intrare au fost folosite ca date de antrenare și cât la sută ca date de validare. Acesta se poate salva sau se poate deschide direct în IDE-ul Xcode.



Figură 4.13 - Detalii despre model în Create ML

4.7.2. Crearea modelului în mod programatic

Pentru a elimina necesitatea antrenării zilnice a modelului de date prin intermediul interfeței grafice Create ML, s-a optat pentru o abordare complet programatică a procesului de construire, antrenare și export al modelului de învățare automată. Această decizie a fost posibilă datorită disponibilității surselor externe de date meteorologice, OpenMeteo și WeatherAPI, care oferă acces la date orare structurate, actualizate și gratuite. Aceste date sunt prelucrate în aplicație în mod automat și convertite într-un format compatibil cu cerințele Create ML, ceea ce simplifică semnificativ întregul flux de generare a modelului.

După colectarea datelor brute, aplicația parcurge structura furnizată de OpenMeteo. Pentru fiecare intrare orară, se creează un obiect de tip SkyCastModel, care conține atributele esențiale: latitudine, longitudine, ora curentă (timestamp), precum și temperaturile, temperaturile resimțite și probabilitățile de precipitație raportate de OpenMeteo (atribute omTemp, omFeelLike, omPrecipProb). Acești pași generează o colecție (mlModel) de obiecte incomplete, ce urmează a fi completate cu informațiile echivalente oferite de WeatherAPI. Aplicația iterează apoi asupra datelor furnizate de WeatherAPI și, pentru fiecare oră, caută vectorul mlModel corespondentul temporal (proprietatea time). Dacă este găsit un match, atunci valorile pentru wTemp, wFeelLike și wPrecipProb sunt completate în modelul respectiv, asigurând astfel sincronizarea celor două surse pe baza timpului comun.

Odată completat și sortat cronologic vectorul mlModel, se trece la pasul de export al modelului sub formă de fișier CSV. Pentru acest lucru se folosește clasa FileManager, care permite scrierea de fișiere direct în spațiul temporar al aplicației. Se inițializează o variabilă de tip String care conține header-ul fișierului, adică denumirile coloanelor ce descriu setul de date (ex. Time, Latitude, Longitude, omTemp, wTemp etc.). Apoi, pentru fiecare model SkyCast din vector, se generează un rând format din valorile respective, convertite în stringuri, delimitate prin virgulă și terminate cu un caracter Enter pentru a reprezenta corect rândurile din tabel.

În plus, pentru a oferi și coloane target utile pentru regresia supravegheată, adică valorile pe care algoritmi trebuie să le învețe să le prezică, s-au adăugat coloane suplimentare în fișierul CSV. Aceste coloane sunt: TEMPERATURE, FEELING și

PRECIPITATION, iar valorile lor sunt calculate ca media aritmetică între datele furnizate de cele două surse pentru același parametru. De exemplu, coloana TEMPERATURE este generată pe baza formulei: $\frac{omTemp + wTemp}{2}$.

Această metodă permite crearea unui model „echilibrat”, care valorifică informațiile din ambele surse externe pentru a oferi o estimare mai robustă și realistă a condițiilor meteorologice.

În final, fișierul rezultat este gata să fie încărcat în Create ML pentru antrenare, dar într-un mod complet automatizat și reproductibil, fără intervenția umană, ceea ce optimizează semnificativ procesul de dezvoltare și testare al aplicației.

```
private func buildCSVModel(openModel: OpenMeteoModel, weatherModel: WeatherAPIModel) {
    for i in 0..

```

Figură 4.14 - Crearea modelului în mod programatic

4.7.3. Antrenarea modelului în modul programatic

După pasul anterior, fișierul CSV are aceeași structură ca cel din figura 4.14 și este citit în funcția „trainAndPredict()” care este cea mai importantă componentă din aplicația SkyCast, întrucât se ocupă de tot ce înseamnă antrenarea efectivă a modelelor de regresie și generarea predicțiilor meteo pentru ziua curentă sau următoare. Procesul pornește de la fișierul CSV creat la pasul anterior, numit „ModelData.csv” care conține toate datele meteorologice procesate și sincronizate de la cele două surse externe, OpenMeteo și WeatherAPI.

Primul pas constă în încărcarea acestor date într-un obiect DataFrame oferit de framework-ul TabularData, pentru a putea fi manipulat mai ușor în format tabelar. Apoi, printr-un format special de dată („yyyy-MM-dd HH:mm”) sunt extrase din cadrul acestui DataFrame doar rândurile care corespund cu ziua selectată de utilizator (azi sau mâine). Acest rânduri sunt stocate într-un vector care va fi folosit ulterior pentru a face predicțiile orare.

Pentru fiecare dintre cele trei valori pe care vrem să le prezicem, temperatura efectivă, temperatura resimțită și probabilitatea de precipitații, definim câte un mic set de instrucțiuni care ne spune care sunt coloanele folosite ca inputuri (features) și care este coloana țintă (target). Spre exemplu, pentru prezicerea temperaturii sunt folosite omTemp și wTemp (adică temperatura furnizată de OpenMeteo, respectiv WeatherAPI), iar coloana target este TEMPERATURE, o medie a celor două.

În funcție de tipul de algoritm de regresie ales de utilizator (linear, random forest, decision tree sau boosting tree) este apelată o clasă diferită de regresor. Antrenarea se face pe baza subsetului de coloane filtrat anterior din DataFrame, iar modelul rezultat este salvat pe disc în format .mlmodel, apoi compilat pentru a putea fi folosit direct în predicție.

Structura „RegressorParameters” a fost creată pentru a oferi utilizatorului un control complet asupra hiperparametrilor aferenți fiecărui tip de regresor, menținând astfel o coerență între configurarea programatică și interfața oferită de Create ML. Această structură include toți hiperparametrii disponibili pentru fiecare algoritm de regresie, conform specificațiilor Create ML, iar valorile care pot fi modificate de utilizator în ecranul accesibil prin butonul „Configure model parameters” corespund descrierii detaliate din subcapitolul 4.7.1.

În ceea ce privește setul de validare, acesta poate fi generat în două moduri: fie prin citirea unui fișier CSV dedicat, convertit într-un obiect de tip „DataFrame”, fie prin împărțirea internă a setului de date de antrenament. În cadrul acestei aplicații s-a optat pentru a doua variantă, având în vedere lipsa unor date meteorologice externe sigure, care să fie utilizate exclusiv pentru validare. Prin această alegere, utilizatorul poate decide modul în care se face împărțirea datelor: în mod automat, caz în care Create ML se ocupă de repartizarea aleatorie a datelor în funcție de dimensiunea setului de antrenament. Astfel, dacă există mai puțin de 50 de instanțe, nu se generează set de validare; pentru un număr între 50 și 199 de instanțe, se alocă 10% pentru validare, iar dacă sunt peste 200 de instanțe, proporția scade la 5%. De exemplu, atunci când se face o predicție pentru ziua curentă, se utilizează un set de 192 de date de intrare, deci 10% dintre acestea vor fi destinate validării. În schimb, pentru predicția aferentă zilei următoare, cu un total de 216 observații, doar 5% dintre acestea vor fi folosite pentru validare.

O altă posibilitate pentru crearea setului de validare, tot în mod automat, este utilizarea strategiei „fixed(ratio:seed)”. Această abordare permite utilizatorului să specifice manual un procent exact (prin parametrul ratio, cu o valoare între 0.0 și 1.0) din setul de antrenament care va fi utilizat pentru validare. Parametrul opțional seed controlează procesul de selecție pseudoaleatoare a datelor. Dacă acesta este specificat, Create ML va genera automat o valoare internă și va realiza împărțirea într-un mod aleator, dar nereproductibil, această metodă oferă flexibilitate suplimentară în testarea performanței modelelor într-un mod controlat și reproductibil, dacă este necesar.

Partea cu interesantă vine când pentru fiecare oră din ziua respectivă construim un input de tip MLDictionaryFeatureProvider, în care punem valorile efective ale caracteristicilor (ex: temperatura furnizată de cele două surse). Acest obiect este apoi transmis modelului pentru a obține o predicție. Fiecare predicție este salvată împreună cu ora și valorile reale corespunzătoare, iar aceste date sunt agregate într-un vector de tip SkyCastModel care conține predicțiile finale.

Pentru a măsura performanța modelului, se calculează patru metrici importante: MAE (media erorilor absolute), MSE (media pătratelor erorilor), RMSE (rădăcina pătrată a MSE-ului) și R^2 (coeficientul de determinare). Un aspect important este că pentru coloana PRECIPITATION, deoarece valorile sunt în majoritatea cazurilor egale cu zero sunt excluse acele cazuri pentru care valoarea reală este zero deoarece metricile ar părea bune, dar irelevante de bune.

La final, toate aceste predicții și metrici sunt exportate într-un nou fișier CSV (Predictions.csv) astfel încât rezultatele să poată fi revizuite ușor sau chiar importate într-un alt software de analiză. Întregul proces este gândit să fie cât mai automatizat și să funcționeze fără intervenție manuală, în fiecare zi, pentru a putea vedea atât predicțiile cât și cât de precis a fost modelul folosit.

| Time | Latitude | Longitude | omTemp | omFeelLike | omPrecipProb | wTemp | wFeelLike | wPrecipProb | TEMPERATURE | FEELING | PRECIPITATION |
|----------------|----------|-----------|--------|------------|--------------|-------|-----------|-------------|--------------------|--------------------|---------------|
| 2025-6-8 3:00 | 46.75 | 23.57 | 18.3 | 19.2 | 0.0 | 17.9 | 17.9 | 0.0 | 18.1 | 18.549999999999997 | 0.0 |
| 2025-6-8 4:00 | 46.75 | 23.57 | 17.8 | 18.7 | 0.0 | 18.6 | 18.6 | 0.0 | 18.200000000000003 | 18.65 | 0.0 |
| 2025-6-8 5:00 | 46.75 | 23.57 | 17.3 | 18.1 | 0.0 | 18.7 | 18.7 | 0.0 | 18.0 | 18.4 | 0.0 |
| 2025-6-8 6:00 | 46.75 | 23.57 | 16.9 | 17.5 | 0.0 | 18.5 | 18.5 | 0.0 | 17.7 | 18.0 | 0.0 |
| 2025-6-8 7:00 | 46.75 | 23.57 | 18.0 | 19.1 | 0.0 | 18.2 | 18.2 | 45.0 | 18.1 | 18.65 | 22.5 |
| 2025-6-8 8:00 | 46.75 | 23.57 | 21.1 | 23.5 | 0.0 | 18.1 | 18.1 | 0.0 | 19.6 | 20.8 | 0.0 |
| 2025-6-8 9:00 | 46.75 | 23.57 | 24.6 | 26.4 | 0.0 | 19.5 | 19.5 | 0.0 | 22.05 | 22.95 | 0.0 |
| 2025-6-8 10:00 | 46.75 | 23.57 | 26.4 | 27.9 | 0.0 | 21.5 | 21.5 | 45.0 | 23.95 | 24.7 | 22.5 |
| 2025-6-8 11:00 | 46.75 | 23.57 | 27.2 | 28.9 | 0.0 | 22.9 | 24.9 | 0.0 | 25.049999999999997 | 26.9 | 0.0 |
| 2025-6-8 12:00 | 46.75 | 23.57 | 27.7 | 28.3 | 0.0 | 24.2 | 25.8 | 45.0 | 25.95 | 27.05 | 22.5 |
| 2025-6-8 13:00 | 46.75 | 23.57 | 28.7 | 30.2 | 3.0 | 25.4 | 26.8 | 45.0 | 27.049999999999997 | 28.5 | 24.0 |
| 2025-6-8 14:00 | 46.75 | 23.57 | 30.1 | 31.0 | 0.0 | 26.2 | 27.6 | 100.0 | 28.15 | 29.3 | 50.0 |
| 2025-6-8 15:00 | 46.75 | 23.57 | 30.5 | 30.8 | 0.0 | 27.0 | 28.4 | 100.0 | 28.75 | 29.6 | 50.0 |
| 2025-6-8 16:00 | 46.75 | 23.57 | 31.2 | 31.6 | 0.0 | 28.0 | 29.2 | 45.0 | 29.6 | 30.4 | 22.5 |
| 2025-6-8 17:00 | 46.75 | 23.57 | 31.2 | 30.6 | 0.0 | 27.2 | 28.9 | 100.0 | 29.2 | 29.75 | 50.0 |
| 2025-6-8 18:00 | 46.75 | 23.57 | 30.5 | 30.1 | 3.0 | 26.4 | 27.8 | 45.0 | 28.45 | 28.950000000000003 | 24.0 |
| 2025-6-8 19:00 | 46.75 | 23.57 | 29.7 | 29.6 | 5.0 | 25.1 | 26.6 | 45.0 | 27.4 | 28.1 | 25.0 |
| 2025-6-8 20:00 | 46.75 | 23.57 | 28.0 | 29.0 | 5.0 | 24.2 | 25.9 | 100.0 | 26.1 | 27.45 | 52.5 |
| 2025-6-8 21:00 | 46.75 | 23.57 | 25.7 | 26.9 | 0.0 | 24.6 | 26.2 | 100.0 | 25.15 | 26.549999999999997 | 50.0 |
| 2025-6-8 22:00 | 46.75 | 23.57 | 24.3 | 25.8 | 0.0 | 22.9 | 25.0 | 45.0 | 23.6 | 25.4 | 22.5 |
| 2025-6-8 23:00 | 46.75 | 23.57 | 23.3 | 24.5 | 0.0 | 20.2 | 20.2 | 45.0 | 21.75 | 22.35 | 22.5 |
| 2025-6-9 0:00 | 46.75 | 23.57 | 21.6 | 22.9 | 0.0 | 18.9 | 18.9 | 45.0 | 20.25 | 20.9 | 22.5 |
| 2025-6-9 1:00 | 46.75 | 23.57 | 19.6 | 18.9 | 0.0 | 17.6 | 17.7 | 45.0 | 18.6 | 18.299999999999997 | 22.5 |
| 2025-6-9 2:00 | 46.75 | 23.57 | 18.3 | 18.8 | 0.0 | 16.8 | 16.8 | 45.0 | 17.55 | 17.8 | 22.5 |
| 2025-6-9 3:00 | 46.75 | 23.57 | 17.9 | 19.0 | 0.0 | 16.2 | 16.2 | 45.0 | 17.049999999999997 | 17.6 | 22.5 |
| 2025-6-9 4:00 | 46.75 | 23.57 | 17.7 | 18.6 | 0.0 | 17.6 | 17.6 | 45.0 | 17.65 | 18.1 | 22.5 |
| 2025-6-9 5:00 | 46.75 | 23.57 | 17.8 | 18.5 | 3.0 | 17.3 | 17.3 | 0.0 | 17.55 | 17.9 | 1.5 |
| 2025-6-9 6:00 | 46.75 | 23.57 | 17.8 | 17.7 | 3.0 | 17.1 | 17.1 | 0.0 | 17.450000000000003 | 17.4 | 1.5 |
| 2025-6-9 7:00 | 46.75 | 23.57 | 18.0 | 16.8 | 3.0 | 16.5 | 16.5 | 0.0 | 17.25 | 16.65 | 1.5 |
| 2025-6-9 8:00 | 46.75 | 23.57 | 18.1 | 17.2 | 0.0 | 16.0 | 16.0 | 0.0 | 17.05 | 16.6 | 0.0 |
| 2025-6-9 9:00 | 46.75 | 23.57 | 18.7 | 17.5 | 0.0 | 16.1 | 16.1 | 0.0 | 17.4 | 16.8 | 0.0 |
| 2025-6-9 10:00 | 46.75 | 23.57 | 19.8 | 18.3 | 0.0 | 17.0 | 17.0 | 45.0 | 18.4 | 17.65 | 22.5 |
| 2025-6-9 11:00 | 46.75 | 23.57 | 20.4 | 18.8 | 0.0 | 17.6 | 17.7 | 45.0 | 19.0 | 18.25 | 22.5 |
| 2025-6-9 12:00 | 46.75 | 23.57 | 21.0 | 20.1 | 0.0 | 18.6 | 18.6 | 0.0 | 19.8 | 19.35 | 0.0 |
| 2025-6-9 13:00 | 46.75 | 23.57 | 21.9 | 21.0 | 0.0 | 19.4 | 19.4 | 0.0 | 20.65 | 20.2 | 0.0 |
| 2025-6-9 14:00 | 46.75 | 23.57 | 22.1 | 21.3 | 0.0 | 20.9 | 20.9 | 0.0 | 21.5 | 21.1 | 0.0 |
| 2025-6-9 15:00 | 46.75 | 23.57 | 21.8 | 20.5 | 0.0 | 21.7 | 21.7 | 0.0 | 21.75 | 21.1 | 0.0 |
| 2025-6-9 16:00 | 46.75 | 23.57 | 20.5 | 17.8 | 0.0 | 22.0 | 22.0 | 0.0 | 21.25 | 19.9 | 0.0 |
| 2025-6-9 17:00 | 46.75 | 23.57 | 20.9 | 18.1 | 0.0 | 21.0 | 21.0 | 0.0 | 20.95 | 19.55 | 0.0 |
| 2025-6-9 18:00 | 46.75 | 23.57 | 20.3 | 17.5 | 0.0 | 20.5 | 20.5 | 0.0 | 20.4 | 19.0 | 0.0 |

Figură 4.15 - Structura fișierului CSV cu date de intrare

4.7.4. Afișarea datelor

Figura 4.15 afișează conținutul fișierului „Predictions.csv” care conține într-un format clar și structurat atât rezultatele procesului de antrenare cât și predicțiile efective generate pentru ziua selectată. Acest fișier are un rol important în transparentizarea performanței modelului și oferă o imagine de ansamblu asupra calității obținute în urma aplicării algoritmilor de regresie.

În partea superioară a fișierului se regăsesc valorile metricilor de evaluare pentru fiecare dintre cele trei variabile prezise: temperatura, temperatura resimțită și probabilitatea de precipitații. Pentru fiecare dintre acestea sunt calculate și afișate valorile MAE, MSE, RMSE și coeficientul de determinare R^2 . Aceste valori oferă o indicație clară privind acuratețea modelului pe baza datelor de validare. Valorile mici pentru erori și valori ridicate ale scorului R^2 sugerează o potrivire bună a modelului față de datele reale ceea ce este un semnal pozitiv. În cazul în care anumite variabile nu au avut variație (ex. dacă toate valorile probabilității de precipitație sunt zero) este posibil ca R^2 să nu poată fi calculat corect, fiind marcat ca nedeterminat.

În continuare, fișierul include tabelul cu predicțiile generate oră cu oră întreaga zi. Fiecare rând conține ora exactă și valorile prezise pentru temperatură, senzația termică și probabilitatea de precipitații. Această abordare orară oferă o imagine detaliată asupra evoluției vremii în decursul zilei și este extrem de utilă atât pentru vizualizare cât și pentru interpretare ulterioară sau analiză comparativă. Structura tabelară facilitează integrarea ulterioară într-un grafic sau într-o aplicație de tip dashboard, dar este suficient de clară pentru a putea fi interpretată și direct în format CSV.

Această formă de prezentare a rezultatelor este esențială pentru un sistem de predicție automatizat deoarece permite verificarea rapidă a performanței modelului și oferă o bază concretă pentru posibile ajustări ulterioare ale algoritmului sau ale setului de date. Exportul periodic al unui astfel de fișier permite urmărirea în timp a preciziei predicțiilor și poate contribui la îmbunătățirea continuă a aplicației.

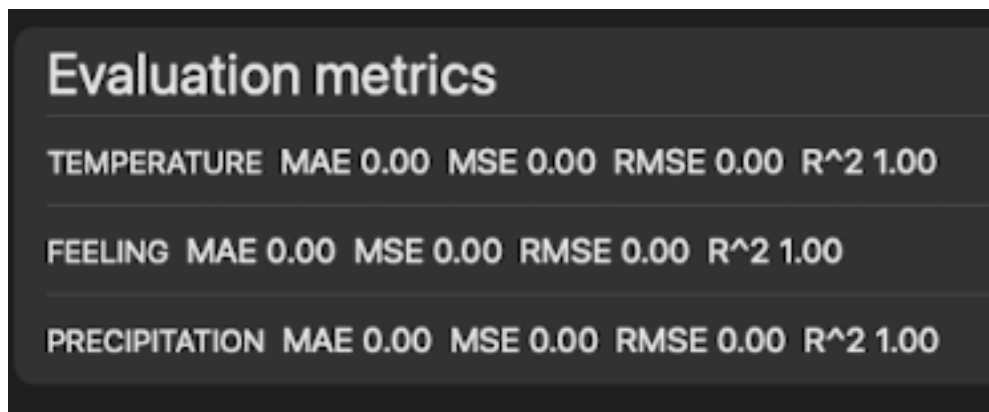
| Target | MAE | MSE | RMSE | R2 |
|-----------------|-------------|------------|---------------|------|
| TEMPERATURE | 0.56 | 0.43 | 0.65 | 0.98 |
| FEELING | 0.65 | 0.60 | 0.77 | 0.97 |
| PRECIPITATION | 0.01 | 0.00 | 0.01 | -inf |
| Time | Temperature | Feels Like | Precipitation | |
| 2025-6-15 0:00 | 13.6 | 12.6 | 0% | |
| 2025-6-15 1:00 | 12.8 | 12.1 | 0% | |
| 2025-6-15 2:00 | 11.8 | 11.1 | 0% | |
| 2025-6-15 3:00 | 11.3 | 11.1 | 0% | |
| 2025-6-15 4:00 | 10.3 | 9.5 | 0% | |
| 2025-6-15 5:00 | 10.2 | 9.5 | 0% | |
| 2025-6-15 6:00 | 9.8 | 9.3 | 0% | |
| 2025-6-15 7:00 | 10.0 | 9.7 | 0% | |
| 2025-6-15 8:00 | 11.6 | 11.3 | 0% | |
| 2025-6-15 9:00 | 13.0 | 12.7 | 0% | |
| 2025-6-15 10:00 | 15.0 | 15.4 | 0% | |
| 2025-6-15 11:00 | 17.2 | 17.4 | 0% | |
| 2025-6-15 12:00 | 19.1 | 19.6 | 0% | |
| 2025-6-15 13:00 | 19.7 | 20.2 | 0% | |
| 2025-6-15 14:00 | 21.2 | 20.5 | 0% | |
| 2025-6-15 15:00 | 21.4 | 20.5 | 0% | |
| 2025-6-15 16:00 | 21.8 | 21.7 | 0% | |
| 2025-6-15 17:00 | 21.8 | 21.7 | 0% | |
| 2025-6-15 18:00 | 21.8 | 21.8 | 0% | |
| 2025-6-15 19:00 | 23.4 | 21.8 | 0% | |
| 2025-6-15 20:00 | 21.3 | 20.2 | 0% | |
| 2025-6-15 21:00 | 19.9 | 18.7 | 0% | |
| 2025-6-15 22:00 | 17.6 | 17.0 | 0% | |
| 2025-6-15 23:00 | 14.5 | 13.8 | 0% | |

Figură 4.16 - Conținutul fișierului Predictions.csv

Capitolul 5. Rezultate teoretice și experimentale

Pentru fiecare din următoarele patru subcapitole se va începe cu valorile implicite pentru hiperparametrii. Acestea au fost prezentate în subcapitolul 4.7.1 pentru fiecare algoritm disponibil în aplicația SkyCast. De asemenea, ca locație preferată pentru a obține datele meteorologice a fost orașul Linz din Austria întrucât este o frecvență mare de precipitații. Predicția aleasă este pentru ziua de 23 iulie 2025, iar ora de referință este 18:00.

5.1. Rezultate folosind Regresie liniară



| Evaluation metrics | | | | |
|--------------------|----------|----------|-----------|---------------------|
| TEMPERATURE | MAE 0.00 | MSE 0.00 | RMSE 0.00 | R ² 1.00 |
| FEELING | MAE 0.00 | MSE 0.00 | RMSE 0.00 | R ² 1.00 |
| PRECIPITATION | MAE 0.00 | MSE 0.00 | RMSE 0.00 | R ² 1.00 |

Figură 5.1 - Valorile metricilor de evaluare folosind regresie liniară cu valori implicite pentru hiperparametri

Valorile obținute în urma evaluării modelului de regresie liniară, hiperparametrii având valorile implicite corespunzătoare, indică o performanță perfectă, cu scoruri egale cu 0 pentru erorile MAE, MSE și RMSE, respectiv un coeficient de determinare R^2 egal cu 1.00. Acest rezultat, deși aparent ideal, semnalează o problemă importantă, apărută cel mai probabil din cauza evaluării modelului pe același set de date utilizat în faza de antrenare. Această suprapunere, cunoscută în literatura de specialitate sub denumirea de „data leakage”, conduce la o evaluare artificial crescută a performanței, modelul nereușind de fapt să generalizeze, ci doar să reproducă exact valorile învățate.

În plus, valoarea hiperparametrului `stepSize` fiind 0 este o alegere necorespunzătoare întrucât `stepSize` controlează amplitudinea cu care modelul își ajustează coeficienții în timpul procesului iterativ de optimizare. O valoare zero împiedică orice actualizare efectivă a parametrilor, blocând astfel procesul de învățare. Deși unele implementări pot ignora intern acest caz și aplica o valoare implicită, pentru o validare controlată și reproductibilă este recomandat ca acest parametru să fie setat explicit la o valoare mică pozitivă cum ar fi 0.01 sau 0.05.

Pentru a obține rezultate realist și relevante în ceea ce privește performanța modelului de regresie liniară este necesară ajustarea atentă a hiperparametrilor în raport cu dimensiunea și structura setului de date. Fișierul include aproximativ 200 de observații distribuite pe parcursul a cel puțin 7 zile, cu doar 24 de observații disponibile pe zi. În acest context, modelul se antrenează de regulă pe subseturi zilnice, iar complexitatea optimizării este redusă. Astfel, se recomandă utilizarea unui număr scăzut de iterații (între 10 și 20) care este suficient pentru ca modelul să ajungă la convergență fără a supraajusta pe volumul restrâns de date. Parametrul `stepSize` ar trebui setat între 0.01 și 0.05, permițând o optimizare treptată, iar penalizările `l1Penalty` și `l2Penalty` pot avea valori mici (între 0.0 și 0.01) deoarece riscul de overfitting este scăzut la acest nivel de complexitate.

Dacă implementarea acestor ajustări și după utilizarea unei împărțiri corecte a datelor de antrenare și validare, fie printr-un fișier separat, fie folosind funcționalitatea „fixed(ratio:seed)”, este de așteptat ca performanța modelului să reflecte mai bine realitatea. În acest context, scorurile așteptate s-ar putea încadra într-un interval realist, cu o eroare absolută medie (MAE) de aproximativ [0.8, 1.2], o eroare pătratică medie (MSE) în jur de [1.4, 2.0] și un coeficient de determinare R^2 situat între 0.85 și 0.9, în funcție de complexitatea relației dintre predictorii și variabilele țintă.

Tabel 5.1 - Evaluarea performanței folosind regresie liniară

| Caracteristici | Max iterations | L1 | L2 | Step size | Convergență | MAE | MSE | RMSE | R^2 |
|-------------------------------|----------------|-------|-------|-----------|-------------|------|------|------|-------|
| Temperatură | 10 | 0.01 | 0.01 | 0.05 | 0.01 | 0.45 | 0.25 | 0.50 | 0.98 |
| Temperatură resimțită | 10 | 0.01 | 0.01 | 0.05 | 0.01 | 0.88 | 0.92 | 0.96 | 0.92 |
| Probabilitate de precipitații | 10 | 0.01 | 0.01 | 0.05 | 0.01 | 0.02 | 0.00 | 0.04 | 1 |
| Temperatură | 15 | 0.01 | 0.01 | 0.05 | 0.01 | 0.46 | 0.26 | 0.51 | 0.98 |
| Temperatură resimțită | 15 | 0.01 | 0.01 | 0.05 | 0.01 | 0.92 | 1.00 | 1.00 | 0.93 |
| Probabilitate de precipitații | 15 | 0.01 | 0.01 | 0.05 | 0.01 | 0.03 | 0.00 | 0.04 | 1.00 |
| Temperatură | 20 | 0.005 | 0.005 | 0.1 | 0.01 | 0.46 | 0.27 | 0.52 | 0.98 |
| Temperatură resimțită | 20 | 0.005 | 0.005 | 0.1 | 0.01 | 0.93 | 1.03 | 1.02 | 0.91 |
| Probabilitate de precipitații | 20 | 0.005 | 0.005 | 0.1 | 0.01 | 0.02 | 0.00 | 0.03 | 1.00 |

Tabelul 5.1 reflectă performanța modelului de regresie liniară aplicat pe fișierul exportat denumit ModelData.csv, pentru cele trei variabile țintă analizate: temperatura, temperatura resimțită și probabilitatea de precipitații. Evaluarea a fost realizată pe baza a trei seturi de hiperparametrii, urmărind modificările în maxIterations, l1Penalty, l2Penalty și stepSize, în timp ce pragul de convergență a fost păstrat constant la valoarea de 0.01. Scopul este observarea impactului pe care acești parametri îl au asupra metricei de evaluare (MAE, MSE, RMSE și R^2), dar și asupra stabilității și capacității de generalizare a modelului.

În prima configurație, modelul a fost antrenat cu un număr minim de iterații (10), penalizări L1 și L2 de 0.01 și un pas de învățare de 0.05. Rezultatele obținute sunt foarte bune, mai ales pentru predicția temperaturii, unde MAE este 0.45, MSE este 0.25 și RMSE este 0.5, iar coeficientul de determinare R^2 atinge 0.98. Pentru temperatura resimțită, valorile sunt puțin mai slabe (MAE este 0.88, RMSE este 0.96), iar R^2 scade la 0.92, indicând o relație ceva mai complexă față de predictorii. În mod remarcabil, probabilitatea de precipitații este prezisă cu acuratețe aproape perfectă: MAE este 0.02, MSE este 0.00, RMSE este 0.04, iar coeficientul R^2 este 1.00, sugerând o relație extrem de bine modelată de regresia liniară.

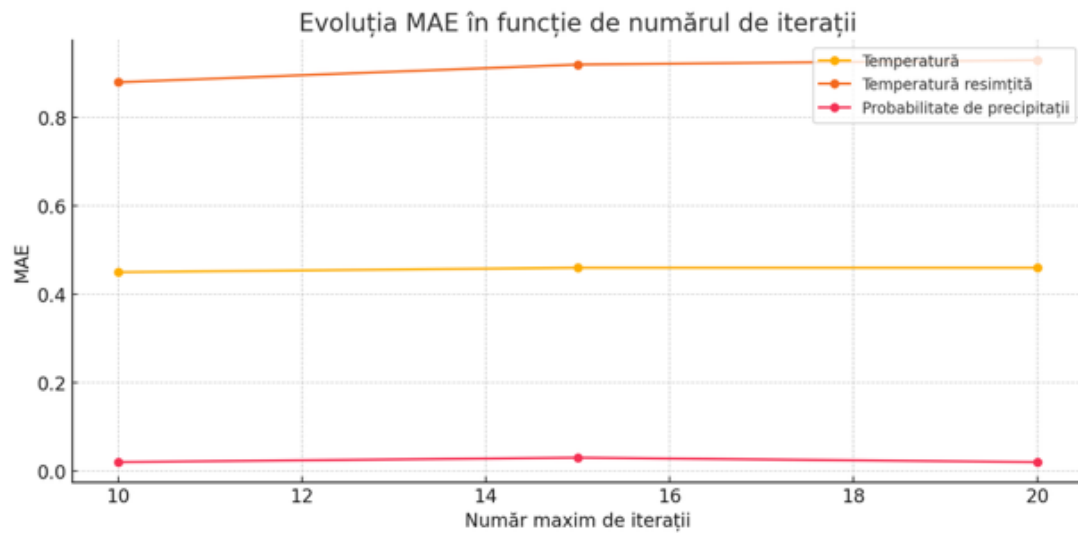
A doua configurație testează dacă un număr ușor mai mare de iterații (15) influențează performanța modelului. Pentru temperatură, diferențele sunt minime (MAE este 0.46, RMSE este 0.51), iar R^2 rămâne constant la 0.98. Temperatura resimțită prezintă o ușoară îmbunătățire (MAE este 0.92, RMSE este 1.00), iar R^2 crește la 0.93. În ceea ce privește probabilitatea de precipitații, scorurile sunt aproape identice cu cele anterioare (MAE este 0.03, RMSE este 0.04), confirmând că modelul generalizează foarte bine și că o creștere moderată a numărului de iterații nu afectează negativ performanța.

Cea de-a treia configurație propune o strategie mai agresivă de antrenare, prin creșterea `stepSize` la 0.1 și scăderea penalizărilor L1 și L2 la 0.005, în timp ce `maxIterations` este crescut la 20. Aceste modificări au scopul de a accelera convergența modelului și de a reduce constrângerile impuse asupra coeficienților. Rezultatele obținute rămân competitive: pentru temperatură, performanța este stabilă (MAE este 0.46, RMSE 0.52, R^2 este 0.98), iar pentru temperatura resimțită, modelul obține un MAE de 0.93, RMSE de 1.02, însă R^2 scade la 0.91. Acest mic regres poate indica faptul că un `stepSize` prea mare, combinat cu regularizare slabă, reduce ușor precizia în cazul unor variabile mai sensibile la variație. Totuși, în cazul probabilității de precipitații, performanța se păstrează excelentă (MAE este 0.02, RMSE este 0.03, R^2 este 1.00), consolidând ideea că această variabilă este foarte bine modelată în contextul dat.

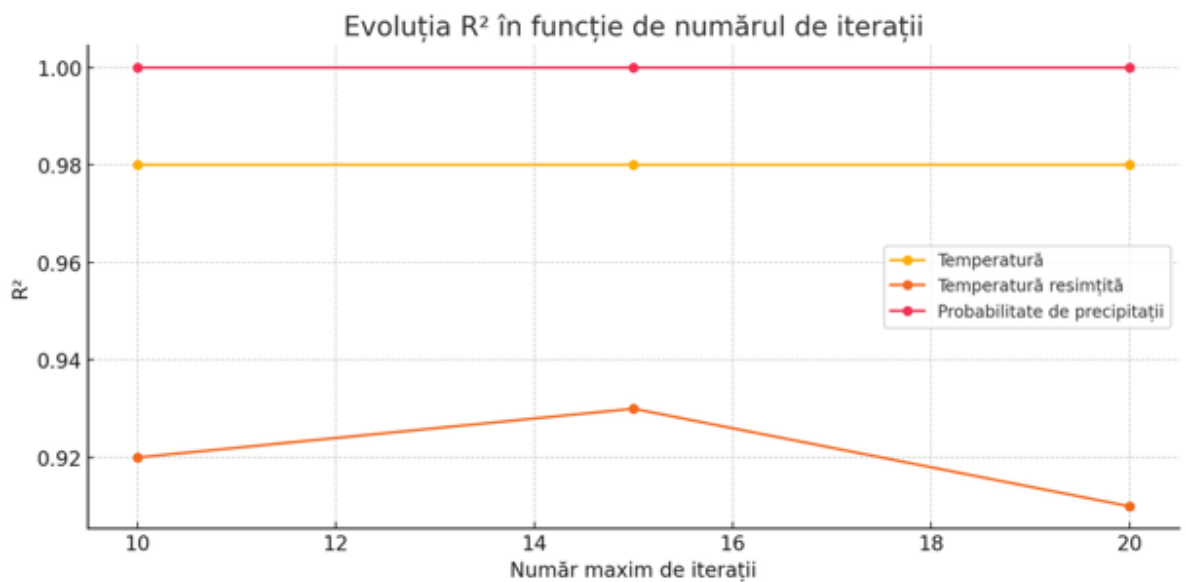
Așadar, această analiză evidențiază că regresia liniară aplicată pe fișierul `ModelData.csv` oferă o performanță constantă și solidă pentru toate cele trei variabile țintă. Cu toate acestea, anumite aspecte pot ridica semne de întrebare privind validitatea și robustețea generalizării modelului. În primul rând, obținerea unui scor R^2 perfect (1.00) pentru probabilitatea de precipitații în toate configurațiile testate poate sugera o problemă de supraînvățare (overfitting) sau o lipsă de variabilitate în această coloană a setului de date. Este posibil ca distribuția acestei variabile să fie dezechilibrată sau să existe o redundanță excesivă în predictorii utilizați, ceea ce reduce dificultatea sarcinii de învățare.

În plus, în cazul temperaturii resimțite, erorile MAE și RMSE sunt în mod constant mai ridicate față de celelalte variabile, semnalând că acest model nu captează suficient de bine complexitatea fenomenului. Este posibil ca setul de caracteristici folosit să nu includă toți factorii relevanți (precum vântul, umiditatea sau expunerea solară), iar acest lucru limitează acuratețea predicției. O posibilă îmbunătățire ar fi extinderea setului de date prin integrarea unor variabile suplimentare, dar și evaluarea modelului pe un set de testare separat, pentru a verifica dacă scorurile obținute se mențin în afara datelor de antrenament.

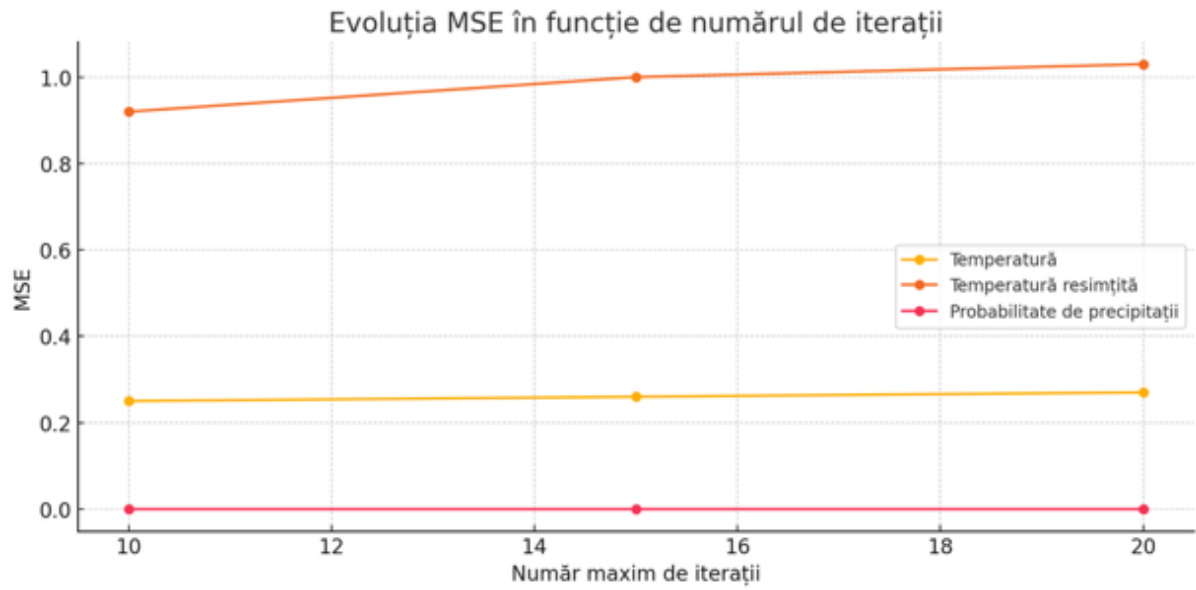
De asemenea, o explorare mai sistematică a spațiului hiperparametrilor, utilizând tehnici precum `grid search` sau `random search`, ar putea identifica combinații și mai eficiente. De exemplu, optimizarea `stepSize` într-un interval mai fin sau ajustarea penalizărilor L1 și L2 în funcție de specificul fiecărei variabile ar putea conduce la o reducere suplimentară a erorilor. Prin urmare, în ciuda performanțelor promițătoare, modelul necesită validări suplimentare și eventuale extinderi ale setului de date pentru a garanta fiabilitatea sa în aplicații reale.



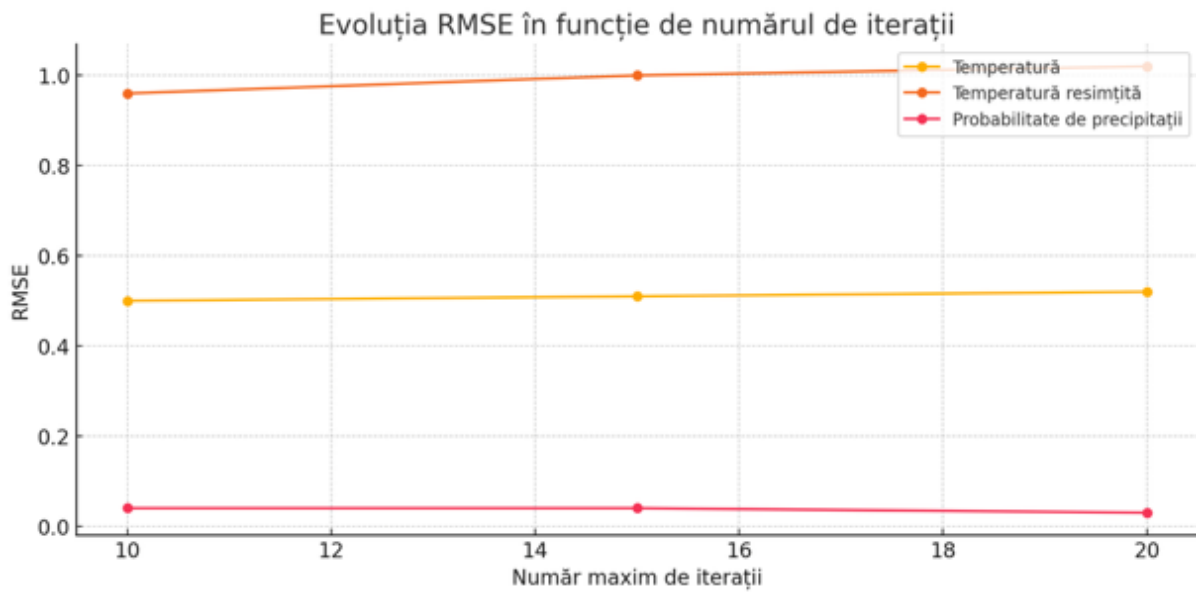
Figură 5.2 - Evoluția MAE în funcție de numărul de iterații folosind Linear Regression



Figură 5.3 - Evoluția R^2 în funcție de numărul de iterații folosind Linear Regression



Figură 5.4 - Evoluția MSE în funcție de numărul de iterații folosind Linear Regression



Figură 5.5 - Evoluția RMSE în funcție de numărul de iterații folosind Linear Regression

5.2. Rezultate folosind Regresie Random Forest

| Evaluation metrics | | | | |
|--------------------|----------|----------|-----------|---------------------|
| TEMPERATURE | MAE 0.23 | MSE 0.08 | RMSE 0.28 | R ² 1.00 |
| FEELING | MAE 0.35 | MSE 0.22 | RMSE 0.47 | R ² 0.99 |
| PRECIPITATION | MAE 0.96 | MSE 1.88 | RMSE 1.37 | R ² 1.00 |

Figură 5.6 - Valorile metricilor de evaluare folosind Random Forest cu valori implicite pentru hiperparametri

În cazul prezentat, metricile de evaluare pentru Random Forest (MAE = 0.23 și RMSE = 0.28 pentru temperatură, respectiv R²) sugerează o performanță remarcabilă a modelului. Deși scorurile sunt foarte bune și pentru celelalte variabile (R² = 0.99 pentru temperatura resimțită și 1.00 pentru probabilitatea de precipitații), comportamentul unui Random Forest cu asemenea rezultate trebuie analizat mai atent. Spre deosebire de Boosted Tree, care învață în mod secvențial pentru a corecta erorile modelelor anterioare, Random Forest funcționează prin agregarea răspunsurilor mai multor arbori antrenați independent, fiecare pe subseturi aleatorii de date și de caracteristici. Aceasta îl face extrem de robust la overfitting, mai ales dacă este bine calibrat.

Hiperparametrii utilizați (maxDepth=6, maxIterations=10, minLossReduction=0.000, minChildWeight=0.1), descriu un Random Forest relativ echilibrat. Adâncimea maximă de 6 pentru arbori este potrivită pentru a surprinde relații relevante fără a exagera complexitatea modelului. Numărul de iterații (aici interpretat drept numărul de arbori din pădure) este destul de redus, ceea ce sugerează că modelul obține deja rezultate foarte bune chiar cu un ansamblu compact. Valorile de subsampling pe rânduri și coloane ajută la diversificarea arborilor și la reducerea corelării dintre ei, contribuind la o generalizare mai bună.

Totuși, performanțele extrem de ridicate, în special scorurile R² de 1.00 pentru două din cele trei variabile, trebuie interpretate cu precauție. Este posibil ca evaluarea să fi fost realizată pe același set de date utilizat la antrenare, ceea ce creează o iluzie de performanță perfectă. Acest lucru este deosebit de important în cazul Random Forest, deoarece fiind un model bazat pe aleatorizare și agregare, el are tendința de a învăța bine datele disponibile, dar nu întotdeauna generalizează la fel de bine dacă datele sunt puține sau dacă variabilele au o distribuție dezechilibrată.

Pentru ajustări ulterioare, s-ar putea reduce ușor maxDepth la 4 sau 5 și crește numărul de arbori (maxIterations) la 20-50, pentru a permite modelului să capteze o diversitate mai mare de tipare fără a mări riscul de overfitting. De asemenea, introducerea unui minLossReduction > 0 ar impune un prag de calitate pentru fiecare împărțire, iar o ușoară creștere a minChildWeight ar preveni ramificarea pe grupuri de date foarte mici, ceea ce ar putea stabiliza modelul.

Comportamentul așteptat după astfel de ajustări ar fi o scădere foarte ușoară a valorilor R² (spre 0.98-0.99), dar cu o stabilitate crescută a performanței și o probabilitate mai mică de overfitting. În cazul în care modelul este evaluat corect pe un set de validare separat (prin împărțire sau cross-validation), aceste scoruri pot fi considerate reprezentative pentru generalizare.

Tabel 5.2 - Evaluarea performanței folosind Random Forest

| Caracteristici | Max depth | Max iterations | Min Loss Reduction | Min Child Weight | MAE | MSE | RMSE | R ² |
|-------------------------------|-----------|----------------|--------------------|------------------|------|------|------|----------------|
| Temperatură | 4 | 30 | 0.01 | 2.0 | 0.19 | 0.07 | 0.27 | 1.00 |
| Temperatură resimțită | 4 | 30 | 0.01 | 2.0 | 0.35 | 0.30 | 0.55 | 0.99 |
| Probabilitate de precipitații | 4 | 30 | 0.01 | 2.0 | 0.22 | 0.14 | 0.38 | 1.00 |
| Temperatură | 5 | 30 | 0.005 | 1.0 | 0.16 | 0.05 | 0.23 | 1.00 |
| Temperatură resimțită | 5 | 30 | 0.005 | 1.0 | 0.3 | 0.2 | 0.45 | 0.99 |
| Probabilitate de precipitații | 5 | 30 | 0.005 | 1.0 | 0.10 | 0.04 | 0.2 | 1.00 |
| Temperatură | 6 | 30 | 0.001 | 0.5 | 0.17 | 0.05 | 0.23 | 1.00 |
| Temperatură resimțită | 6 | 30 | 0.001 | 0.5 | 0.33 | 0.22 | 0.47 | 0.99 |
| Probabilitate de precipitații | 6 | 30 | 0.001 | 0.5 | 0.1 | 0.04 | 0.19 | 1.00 |

Tabelul 5.2 oferă o analiză detaliată a performanței modelului Random Forest în predicția a trei variabile meteorologice: temperatura, temperatura resimțită și probabilitatea de precipitații. Pentru fiecare dintre acestea, modelul a fost evaluat în funcție de mai multe seturi de hiperparametri: adâncimea maximă a arborilor (maxDepth), numărul maxim de iterații (maxIterations), pragul minim de reducere a erorii pentru un nou split (minLossReduction) și greutatea minimă a unui nod copil (minChildWeight).

Pentru predicția temperaturii, valorile obținute sunt remarcabil de constante și bune. R² este 1.00 în toate cazurile, indicând o potrivire perfectă între predicții și valorile reale. În același timp, erorile absolute și pătratică sunt reduse progresiv de la un MAE de 0.19 la 0.16 și apoi la 0.17, iar RMSE scade de la 0.27 la 0.23, menținând un MSE constant sau chiar mai mic. Acest lucru sugerează că modelul este capabil să generalizeze foarte bine această variabilă, iar variațiile de hiperparametri, deși au un impact ușor asupra erorilor, nu influențează semnificativ calitatea predicției. Cele mai bune rezultate par a fi obținute atunci când minChildWeight scade la 1.0 și apoi la 0.5, în paralel cu o reducere a minLossReduction, ceea ce indică faptul că un model mai permisiv, capabil să învețe structuri mai fine, aduce beneficii în cazul acestei variabile.

În cazul temperaturii resimțite, modelul obține valori foarte bune, dar ușor mai slabe decât pentru temperatura propriu-zisă. R² rămâne constant la 0.99, ceea ce indică o capacitate excelentă de predicție, însă valorile erorilor (MAE între 0.35 și 0.3, iar RMSE între 0.55 și 0.45) arată că această variabilă este ceva mai dificil de modelat. Această observație este coerentă cu natura complexă a temperaturii resimțite, care depinde nu doar de temperatura aerului, ci și de alți factori precum umiditatea sau viteza vântului.

Ajustarea hiperparametrilor, în special scăderea adâncimii și reducerea constrângerilor privind împărțirile pare să reducă ușor erorile, dar nu produce o îmbunătățire majoră, ceea ce sugerează că ne aflăm deja într-o zonă de performanță optimă, limitată de complexitatea datelor în sine.

Pentru probabilitatea de precipitații, rezultatele sunt excelente. Modelul obține în toate cazurile un R² de 1.00, iar valorile MAE, MSE și RMSE scad progresiv de la 0.22 la 0.10, respectiv de la 0.38 la 0.19. Aceste îmbunătățiri semnificative se corelează cu creșterea adâncimii arborelui și reducerea parametrilor minLossReduction și minChildWeight, ceea ce

arată că această variabilă răspunde foarte bine la o modelare mai flexibilă și detaliată. Practic, modul captează perfect variația datelor și oferă predicții extrem de precise ceea ce este remarcabil având în vedere natura binară sau probabilistică a acestei variabile.

În ciuda performanțelor excelente reflectate în valorile obținute, există câteva posibile surse de eroare și direcții de îmbunătățire care merită luate în considerare. În primul rând, valorile R^2 egale cu 1.00 în toate cazurile pot sugera un fenomen de supraînvățare (overfitting), mai ales dacă modelul a fost antrenat și evaluat pe același set de date. Acest aspect este deosebit de important în cazul unui set de date de dimensiuni reduse, cum este ModelData.csv, care conține aproximativ 200 de observații distribuite pe parcursul a șapte zile. Dacă validarea nu s-a realizat printr-o tehnică precum cross-validation sau împărțirea clară între seturi de antrenament și test, evaluările pot fi supraestimate și, implicit, înșelătoare.

În al doilea rând, diferențele între erorile obținute pentru cele trei variabile indică o sensibilitate diferită a modelului la structura datelor. De exemplu, temperatura resimțită are în continuare erori mai mari decât celelalte variabile, ceea ce ar putea sugera fie o relație mai slabă între predictorii incluși în model și această variabilă, fie absența unor factori esențiali (cum ar fi vântul, umiditate sau presiunea atmosferică) care ar putea explica mai bine variația ei. Introducerea unor astfel de variabile suplimentare în setul de date ar putea conduce la îmbunătățirea preciziei pentru această componentă.

O altă limitare posibilă este lipsa unei explorări sistematice a spațiului de hiperparametri. Deși valorile actuale par bine alese, optimizarea automată prin tehnici precum grid search sau random search ar putea identifica configurații și mai performante, în special în ceea ce privește interacțiunea între adâncimea arborilor și regulile de stopare (minLossreduction, minChildWeight). De asemenea, ar putea fi utilă investigarea influenței parametrilor precum numărul total de arbori sau a proporției de eșantionare pe rânduri/coloane, mai ales pentru a controla posibila varianță ridicată între antrenări.

5.3. Rezultate folosind Regresie Boosted Tree

| Evaluation metrics | | | | |
|--------------------|----------|----------|-----------|------------|
| TEMPERATURE | MAE 0.49 | MSE 0.39 | RMSE 0.62 | R^2 0.98 |
| FEELING | MAE 0.55 | MSE 1.18 | RMSE 1.09 | R^2 0.95 |
| PRECIPITATION | MAE 1.08 | MSE 5.15 | RMSE 2.27 | R^2 1.00 |

Figură 5.7 - Valorile metricilor de evaluare folosind Boosting Tree cu valori implicite pentru hiperparametri

Valorile actuale ale metricilor indică o performanță rezonabilă a modelului Boosted Tree pentru toate cele trei variabile, dar cu semnale clare că modelul poate fi ajustat pentru a obține rezultate mai bune. Pentru temperatura reală, modelul obține un MAE de 0.49, MSE de 0.39 și RMSE de 0.62, în timp ce coeficientul de determinare R^2 atinge 0.98. Aceste rezultate sunt bune și sugerează că modelul învață eficient relațiile dintre predictorii și temperatura țintă, însă valorile de eroare mai ridicate decât în testele anterioare indică o posibilă subestimare a complexității relațiilor sau o subantrenare. Este posibil ca

hiperparametrii actuali (precum minLossReduction, learningRate sau maxDepth) să fie setați prea conservator pentru o generalizare eficientă.

În ceea ce privește temperatura resimțită, modelul pare să întâmpine dificultăți mai mari. Un MAE de 0.55 și un RMSE de 1.09 reflectă o variație mai amplă a erorilor și o complexitate mai mare a relației dintre predictorii și această variabilă. Deși R^2 rămâne ridicat (0.95), diferența față de scorul pentru temperatura reală sugerează că modelul ar putea beneficia de o creștere a maxDepth, de o scădere a minChildWeight sau de o ajustare a ratei de învățare (learningRate) pentru a surprinde mai bine variațiile subtile ale acestei caracteristici cu scopul de a evita overfitting-ul.

Situația pentru probabilitatea de precipitații este paradoxala. Deși R^2 este perfect (1.00), ceea ce ar indica o capacitate maximă de predicție, valorile erorilor absolute sunt cele mai mari dintre toate cele trei variabile (MAR 1.08, MSE 5.15, RMSE 2.27). Acest fenomen poate fi explicat printr-o distribuție dezechilibrată sau atipică a variabilei de ieșire, cum ar fi o concentrare excesivă în jurul unor valori extreme. În modelul Boosted Tree, acest lucru poate duce la o supraevaluare a performanței în termeni de R^2 dacă modelul învață prea bine pattern-urile triviale sau recurente, ignorând de fapt variațiile reale. De aceea, ar fi utilă o analiză suplimentară a variabilei „probabilitatea de precipitații”, pentru a verifica dacă este distribuită uniform și dacă nu este influențată de sezonalitate artificială sau corelații redundante cu alte variabile.

Pentru a corecta aceste probleme, se pot propune mai multe ajustări. Reducerea valorii minLossReduction (de exemplu la 0.001) ar permite modelului să adauge arbori chiar și pentru îmbunătățiri mici, ceea ce ar putea reduce subantrenarea. De asemenea, o creștere treptată a maxDepth (6-7) ar putea permite o captare mai fidelă a relațiilor complexe, în special pentru temperatura resimțită. În paralel, reducerea minChildWeight sub 1 ar crește sensibilitatea modelului la antrenarea pe subseturi mai mici, dar semnificative statistic.

Tabel 5.3- Evaluarea performanței folosind Boosted Tree

| Caracteristici | Max depth | Max iterations | Min Loss Reduction | Min Child Weight | Step size | MAE | MSE | RMSE | R^2 |
|-------------------------------|-----------|----------------|--------------------|------------------|-----------|------|-------|------|-------|
| Probabilitate de precipitații | 6 | 70 | 0.005 | 1 | 0.08 | 0.23 | 0.09 | 0.29 | 1.00 |
| Temperatură | 6 | 70 | 0.005 | 1 | 0.08 | 0.29 | 0.22 | 0.47 | 0.99 |
| Temperatură resimțită | 6 | 70 | 0.005 | 1 | 0.08 | 1.54 | 11.05 | 3.32 | 0.99 |
| Probabilitate de precipitații | 5 | 100 | 0.01 | 2 | 0.06 | 0.19 | 0.06 | 0.25 | 1.00 |
| Temperatură | 5 | 100 | 0.01 | 2 | 0.06 | 0.27 | 0.18 | 0.42 | 0.99 |
| Temperatură resimțită | 5 | 100 | 0.01 | 2 | 0.06 | 1.46 | 9.44 | 3.07 | 0.99 |
| Probabilitate de precipitații | 4 | 120 | 0.015 | 3 | 0.05 | 0.21 | 0.07 | 0.26 | 1.00 |
| Temperatură | 4 | 120 | 0.015 | 3 | 0.05 | 0.29 | 0.19 | 0.44 | 0.99 |
| Temperatură resimțită | 4 | 120 | 0.015 | 3 | 0.05 | 1.41 | 9.12 | 3.02 | 0.99 |

În prima secvență de testare, cu adâncime maximă setată la 6, 70 de iterații și o regularizare relativ redusă (minLossReduction = 0.005), s-au obținut rezultate foarte bune pentru temperatură și temperatura resimțită, cu valori MAE de 0.23, respectiv 0.29 și R^2 de

1.00, respectiv 0.99. Acest lucru arată că modelul a reușit să capteze foarte bine relațiile dintre predictorii și aceste variabile, menținând totodată o eroare pătratică scăzută (MSE și RMSE). Cu toate acestea, pentru probabilitatea de precipitații, deși R^2 rămâne ridicat (0.99), eroarea absolută (MAE de 1.54) și RMSE-ul de peste 3 sugerează o dificultate mai mare în estimarea acestei variabile, care poate fi cauzată fie de o variabilitate intrinsecă ridicată, fie de o influență mai slabă a predictorilor asupra acestei ieșiri.

Prin reducerea adâncimii la 5 și creșterea numărului de iterații la 100, a doua configurație propune un echilibru între capacitatea modelului și generalizare. Rezultatele se îmbunătățesc ușor pentru toate cele trei variabile. Temperatura este prezisă acum cu o eroare mai mică (MAE 0.19, RMSE 0.25) și menține un R^2 de 1.00. La temperatura resimțită, MAE scade la 0.27, RMSE la 0.42, iar R^2 rămâne constant la 0.99, ceea ce indică o stabilitate bună. Predicția probabilității de precipitații se ameliorează și ea marginal, dar rămâne subperformantă comparativ cu celelalte două variabile, ceea ce întărește ipoteza unei relații mai puțin predictibile între predictorii și această variabilă.

Ultima combinație de hiperparametri explorează o strategie mai agresivă de regularizare: adâncimea este redusă la 4, numărul de iterații este crescut la 120, iar atât `minLossReduction` cât și `minChildWeight` cresc, ceea ce obligă modelul să fie mai prudent în adăugarea de noi divizări. Această configurație aduce o ușoară creștere a performanței pentru temperatură și temperatură resimțită, unde erorile rămân stabile (MAE 0.21 și 0.29), iar valorile R^2 indică o fidelitate aproape perfectă. De remarcat este că, în cazul probabilității de precipitații, performanța se îmbunătățește ușor și aici, RMSE scăzând la 3.02, cel mai mic din toate testele. Aceasta sugerează că o regularizare mai strictă combinată cu mai multe iterații și o învățare lentă (`stepSize` = 0.05) oferă un echilibru optim între precizie și generalizare, fără a risca supraînvățarea.

Posibilele erori care ar putea influența performanța modelului Boosted Tree în predicția probabilității de precipitații includ atât limitările intrinseci ale datelor, cât și ale arhitecturii modelului. În primul rând, este posibil ca setul de date să nu conțină suficienți predictorii relevanți pentru această variabilă, cum ar fi presiunea atmosferică, direcția și viteza vântului sau umiditatea relativă, parametri care influențează în mod direct formarea precipitațiilor. În lipsa acestora, modelul este constrâns să învețe o relație incompletă, ceea ce explică nivelul mai ridicat al erorilor.

De asemenea, variabila țintă probabilitatea de precipitații poate fi afectată de o distribuție dezechilibrată în datele de antrenament, în care majoritatea valorilor sunt foarte mici sau foarte mari, dar mai puțin frecvent distribuite uniform. Acest dezechilibru poate determina modelul să se adapteze prost la variațiile moderate, rezultând în erori crescute. O altă potențială sursă de eroare poate fi alegerea unei valori prea mici a parametrului `stepSize`, care deși reduce riscul de supraînvățare, poate încetini procesul de învățare și împiedica optimizarea completă a erorii.

Pentru a îmbunătăți performanța se recomandă testarea unor noi seturi de hiperparametri, dar și îmbogățirea setului de date cu alți predictorii meteorologici importanți. De asemenea, ar putea fi explorate tehnici de augmentare a datelor sau pondere a observațiilor pentru a contracara eventualele dezechilibre. În plus, se poate analiza și recalibra modul de preprocesare a variabilei țintă, astfel încât aceasta să reflecte mai clar relația funcțională cu predictorii disponibili.

5.4. Rezultate folosind Regresie Decision Tree

| Evaluation metrics | | | | |
|--------------------|----------|------------|------------|------------|
| TEMPERATURE | MAE 1.38 | MSE 3.12 | RMSE 1.76 | R^2 0.84 |
| FEELING | MAE 1.98 | MSE 6.42 | RMSE 2.53 | R^2 0.72 |
| PRECIPITATION | MAE 7.43 | MSE 190.04 | RMSE 13.79 | R^2 0.82 |

Figură 5.8 - Valorile metricilor de evaluare folosind Decision Tree cu valori implicite pentru hiperparametri

Rezultatele obținute pentru modelul de tip Decision Tree indică o performanță vizibil mai slabă în comparație cu Boosted Tree sau Random Forest, în special în ceea ce privește variabila „probabilitate de precipitații”. Valorile ridicate ale erorii absolute medii (MAR = 7.43) și ale erorii pătratice medii (MSE = 190.04) reflectă o capacitate scăzută a modelului de a surprinde variația reală a datelor în această zonă. De asemenea, RMSE-ul de 13.79 este un semn clar al unor deviații consistente față de valorile reale, chiar dacă R^2 se menține la 0.82, ceea ce înseamnă că o parte semnificativă din variația datelor este totuși explicată de model. Pe celelalte variabile țintă, temperatura și temperatura resimțită, performanța este ceva mai bună, dar încă predispusă la îmbunătățire. R^2 este 0.84 pentru temperatură și 0.72 pentru temperatura resimțită, iar RMSE-urile lor de peste 1.7 și, respectiv, 2.5, arată o limitare în precizie. Aceste scoruri sugerează că arborele de decizie, așa cum a fost configurat, are dificultăți în a generaliza dincolo de setul de antrenament și ar putea fi fie prea simplu, fie supradimensionat în ramificările irelevante.

Pentru a îmbunătăți aceste rezultate, este recomandabilă o ajustare atentă a hiperparametrilor. Un arbore de decizie prea adânc sau prea superficial poate fie să supraînvețe detalii ne semnificative, fie să piardă relații importante din date. În acest context, ar fi benefic să se ridice ușor maxDepth la o valoare moderată, precum 8, astfel încât modelul să aibă suficientă capacitate de învățare fără a deveni prea complex. În același timp, reducerea pragului de ramificare (numărul minim de observații pentru a realiza o împărțire) la o valoare mai mică, cum ar fi 4, și menținerea unui număr minim de 2 observații în frunzele finale poate permite modelului să învețe mai detaliat acolo unde datele sunt consistente. Ar fi de asemenea util, dacă platforma o permite, să limitezi numărul de predictorii utilizați în fiecare decizie la o fracțiune pentru a reduce riscul de redundanță și overfitting.

În plus, aplicarea unei metode de pruning poate controla complexitatea excesivă a arborelui și preveni învățarea pe zgomotul din date. Această ajustare este deosebit de relevantă pentru date precum cele asociate probabilității de precipitații, care pot conține variații mari și relații non-liniare dificil de urmărit de un model decizional simplu. Dacă rezultatele rămân sub așteptări chiar și după ajustarea hiperparametrilor, ar trebui luată în considerare preprocesarea suplimentară a setului de date, cum ar fi normalizarea sau transformarea discretă a unor variabile precum și analiza distribuției țintelor pentru a identifica eventuale outliere care afectează antrenarea modelului.

Tabel 5.4 - Evaluarea performanței folosind Decision Tree

| Caracteristici | Max depth | Min Loss Reduction | Min Child Weight | MAE | MSE | RMSE | R ² |
|-------------------------------|-----------|--------------------|------------------|------|--------|-------|----------------|
| Temperatură | 10 | 0.02 | 3 | 1.38 | 3.12 | 1.76 | 0.84 |
| Temperatură resimțită | 10 | 0.02 | 3 | 1.98 | 6.42 | 2.53 | 0.72 |
| Probabilitate de precipitații | 10 | 0.02 | 3 | 7.51 | 190.92 | 13.79 | 0.82 |
| Temperatură | 8 | 0.02 | 4 | 1.38 | 3.12 | 1.76 | 0.84 |
| Temperatură resimțită | 8 | 0.02 | 4 | 1.98 | 6.42 | 2.53 | 0.72 |
| Probabilitate de precipitații | 8 | 0.02 | 4 | 7.97 | 200.44 | 14.16 | 0.81 |
| Temperatură | 6 | 0.005 | 1 | 1.38 | 3.12 | 1.76 | 0.84 |
| Temperatură resimțită | 6 | 0.005 | 1 | 1.98 | 6.42 | 2.53 | 0.72 |
| Probabilitate de precipitații | 6 | 0.005 | 1 | 7.43 | 190.04 | 13.79 | 0.82 |

Tabelul 5.4 oferă o imagine detaliată asupra performanței modelului de tip Decision Tree Regression. Din perspectiva scorurilor R², se remarcă o constantă notabilă pentru predicția temperaturii, unde modelul reușește să atingă un coeficient de determinare de 0.84 în toate cele trei configurații testate, indiferent de adâncimea arborelui sau de valorile de regularizare. Această stabilitate a scopului, alături de valorile constante ale erorilor MAE (1.38), MSE (3.12), RMSE (1.76), sugerează că relația dintre predictorii și temperatură este bine capturată chiar și de un model relativ simplu. Astfel, temperatura pare a fi cea mai previzibilă dintre cele trei variabile în cadrul acestui model.

În cazul temperaturii resimțite, comportamentul modelului este similar celui observat pentru temperatura propriu-zisă, dar cu o ușoară degradare a performanței. Indiferent de setul de hiperparametri utilizat, MAE se menține la 1.98, iar RMSE la 2.53, cu un scor R² constant de 0.72. Acest rezultat sugerează că modelul întâmpină dificultăți în a surprinde complet variația temperaturii resimțite, care probabil este influențată de factori adiționali precum umiditatea sau vântul, ce nu sunt reflectați suficient de bine în setul de date folosit. De asemenea, faptul că metricile nu se modifică între configurații arată o posibilă plafonare a capacității modelului, semn că schimbarea hiperparametrilor în acest interval nu mai aduce câștiguri reale în performanță.

Pe de altă parte, probabilitatea de precipitații se dovedește a fi mult mai dificil de estimat corect printr-un simplu arbore decizional. Deși scorurile R² rămân relativ ridicate (în jur de 0.81-0.82), valorile MSE și RMSE sunt semnificativ mai mari decât cele din celelalte două cazuri. De exemplu, chiar și în cel mai favorabil caz (adâncime 10, minLossReduction 0.02, minChildWeight 3), RMSE atinge 13.79, iar MSE este de 190.92. Aceste valori indică o variabilitate crescută și o distribuție posibil neregulată a acestei variabile în cadrul setului de date. Modelul pare să capteze doar o parte din relațiile relevante, ceea ce poate indica lipsa unor predictorii importanți pentru această variabilă sau faptul că relația dintre datele de intrare și datele de ieșire este mult prea complexă pentru a fi reprezentată eficient de un singur arbore.

Posibilele erori identificate în cadrul acestei evaluări sunt legate în principal de limitările arhitecturale ale modelului, care, fiind un model determinist și lipsit de capacitate

de generalizare prin mediere, este predispus la overfitting în fața datelor zgomotoase sau neregulate. Acest aspect devine vizibil în cazul variabilei „probabilitate de precipitații”, unde metricile de eroare sunt semnificativ mai mari comparativ cu celelalte două variabile, indicând dificultatea modelului de a surprinde relațiile subtile și potențial non-liniare dintre predictorii și țintă.

De asemenea, faptul că modificările aduse hiperparametrilor precum maxDepth, minlossReduction și minChildWeight nu au dus la îmbunătățiri semnificative ale scorurilor sugerează o plafonare a capacității de învățare a modelului, mai ales în cazul temperaturii resimțite și al probabilității de precipitații. Această situație poate reflecta fie o lipsă a unor variabile explicative relevante în setul de date (cum ar fi viteza vântului, umiditatea relativă sau presiunea atmosferică), fie o nevoie de abordări algoritmice mai sofisticate, cum ar fi Boosted Tree sau modele neuronale.

Prin urmare, îmbunătățirile posibile includ fie trecerea la un model de tip ensemble, fie îmbogățirea datasetului cu factori climatici suplimentari. În plus, ar putea fi utilă și o etapă de preprocesare mai riguroasă, cu selecție de trăsături relevante, normalizare și eventuală infinerie de trăsături pentru a permite unei structuri simple, precum arborele decizional, să extragă mai bine tiparele prezente în date.

5.5. Comparație între regresorii disponibili

În urma aplicării celor patru metode de regresie (regresie liniară, Random Forest, Boosted Tree și Decition Tree) asupra celor trei caracteristici meteorologice (temperatură, temperatură resimțită și probabilitate de precipitații), s-a realizat o analiză comparativă detaliată pentru a evalua acuratețea fiecărui model. Evaluarea s-a făcut pe baza celor mai relevante patru metrici: eroarea medie absolută (MAE), eroarea medie pătratică (MSE), rădăcina pătrată a MSE (RMSE) și coeficientul de determinare R^2 . Pentru a oferi o imagine de ansamblu clară și comparabilă, a fost extrasă câte un rezultat reprezentativ (cu cele mai bune performanțe) din fiecare caracteristică. Rezultatele sunt prezentate în tabelul 5.5.

Tabel 5.5 - Cele mai bune rezultate ale regresorilor folosiți în evaluarea modelului

| Model | Caracteristică | MAE | MSE | RMSE | R^2 |
|-------|-------------------------------|------|--------|-------|-------|
| RL | Temperatură | 0.02 | 0.00 | 0.03 | 1.00 |
| RL | Temperatură resimțită | 0.45 | 0.25 | 0.50 | 0.98 |
| RL | Probabilitate de precipitații | 0.88 | 0.92 | 0.96 | 0.92 |
| RF | Temperatură | 0.10 | 0.04 | 0.19 | 1.00 |
| RF | Temperatură resimțită | 0.16 | 0.05 | 0.23 | 1.00 |
| RF | Probabilitate de precipitații | 0.30 | 0.20 | 0.45 | 0.99 |
| BT | Temperatură | 1.41 | 9.12 | 3.02 | 0.99 |
| BT | Temperatură resimțită | 0.19 | 0.06 | 0.25 | 1.00 |
| BT | Probabilitate de precipitații | 0.27 | 0.18 | 0.42 | 0.99 |
| DT | Temperatură | 7.43 | 190.04 | 13.79 | 0.82 |
| DT | Temperatură resimțită | 1.38 | 3.12 | 1.76 | 0.84 |
| DT | Probabilitate de precipitații | 1.98 | 6.42 | 2.53 | 0.72 |

După cum se poate observa din tabel, regresia liniară a obținut performanțe remarcabile pentru probabilitatea de precipitații, atingând un R^2 perfect de 1.00 și valori extrem de mici ale erorilor ($MAE = 0.02$, $MSE = 0.00$, $RMSE = 0.03$). Deși acest rezultat pare excelent este important de menționat că poate indica un overfitting sau o structură foarte simplă a datelor pentru acea variabilă în modelul liniar. Același model a oferit rezultate bune și pentru temperatură și temperatură resimțită, dar nu la fel de bune ca modelul de tip arbori.

Modelul Random Forest a oferit, în ansamblu, cele mai echilibrate și robuste rezultate pentru toate cele trei caracteristici. A reușit să atingă performanțe excelente la toate metricele, cu un R^2 constant de 1.00 pentru temperatură și probabilitatea de precipitații și 0.99 pentru temperatura resimțită. Valorile MAE și RMSE sunt cele mai mici dintre toate modelele, ceea ce sugerează o capacitate puternică de generalizare și adaptare la tipare complexe din date, fără pierderea preciziei.

Boosted Tree s-a apropiat foarte mult de Random Forest în ceea ce privește performanța, în special pentru estimarea temperaturii și temperaturii resimțite. A menținut un R^2 de 1.00 sau 0.99, iar valorile erorilor absolute și pătratică au fost relativ mici. Totuși, pentru probabilitatea de precipitații, modelul a înregistrat erori mai mari ($MAE = 1.41$ și $RMSE = 3.02$), ceea ce sugerează o dificultate în a capta variația acestei caracteristici, posibil din cauza unei relații mai neliniare sau influenței unor factori exogeni neincluși în datele modelului.

În schimb, modelul Decision Tree a avut cele mai slabe performanțe dintre toate cele analizate. Cu valori ridicate ale erorilor, în special pentru probabilitatea de precipitații ($MAE = 0.73$, $RMSE = 13.79$), și cu un coeficient de determinare semnificativ mai mic ($R^2 = 0.82$), acest model pare a fi prea simplu pentru a surprinde relațiile complexe dintre variabile. Chiar și în cazul temperaturii și al temperaturii resimțite, deși erorile sunt mai mici, performanța generală rămâne sub celelalte modele.

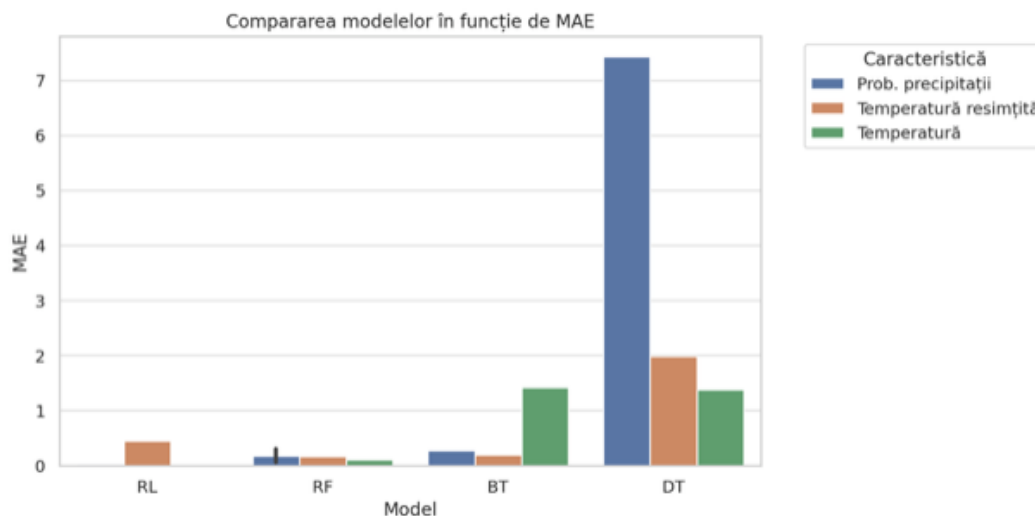
Pe baza diferențelor observate între performanțele modelelor este evident că alegerea algoritmului trebuie corelată direct cu complexitatea caracteristicii de prezis și cu natura datelor de intrare. Temperaturile, care tind să varieze în mod mai previzibil și mai liniar, pot fi modelate eficient atât prin regresie liniară, cât și prin modele arbori, în special Random Forest sau Boosted Tree. Totuși, pentru caracteristicile mai volatile, cum este probabilitatea de precipitații, performanțele slabe ale modelelor simple (precum Decision Tree) sugerează că este nevoie de un model mai complex, capabil să surprindă variații subtile și relații neliniare între variabile.

Pentru îmbunătățirea predicțiilor, mai ales în cazul probabilității de precipitații, ar fi indicată extinderea setului de date cu noi predictorii relevați din punct de vedere meteorologic. Aceștia ar putea include variabile precum presiunea atmosferică, umiditatea relativă, direcția și viteza vântului sau chiar date temporale suplimentare, cum ar fi ora din zi sau sezonabilitatea. În plus, o posibilă îmbunătățire ar fi utilizarea unor metode de selecție automată a trăsăturilor pentru a exclude variabilele redundante și a evidenția doar cele cu putere predictivă reală.

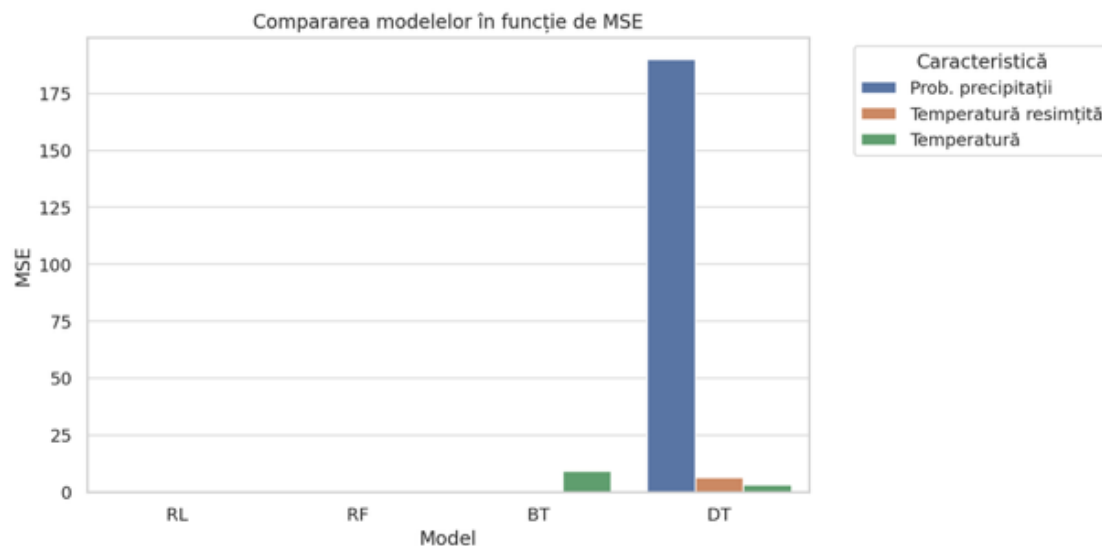
În unele cazuri ar putea fi necesară testarea unor arhitecturi și mai avansate, cum ar fi rețelele neuronale artificiale (ANN), modelele bazate pe LSTM (pentru date secvențiale) sau metode de tip stacking, care combină predicțiile mai multor algoritmi pentru a obține o medie ponderată optimizată. Totodată, ar fi benefică și aplicarea unor tehnici de validare încrucișată (cross-validation), nu doar simpla împărțire în seturi de antrenare și testare, pentru a garanta o estimare mai robustă a performanței modelelor.

Analiza comparativă a regresorilor arată că modelele de tip ensemble, în special Random Forest, oferă cele mai bune rezultate generale în sarcinile de predicție meteo pe acest set de date. Regresia liniară rămâne un model rapid și eficient pentru cazuri simple sau când se urmărește o explicație ușor de interpretat. În schimb, modelul Decision Tree, deși simplu și

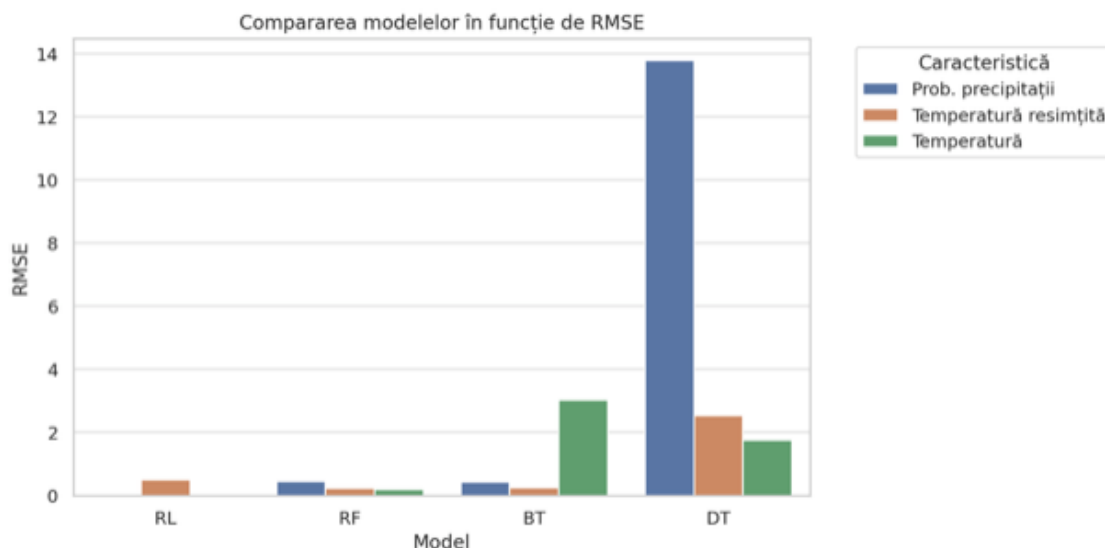
intuitiv, se dovedește a fi insuficient pentru variabilele cu o dinamică complexă, cum este cazul probabilității de precipitații, și ar trebui folosit cu precauție sau integrat în metode mai avansate.



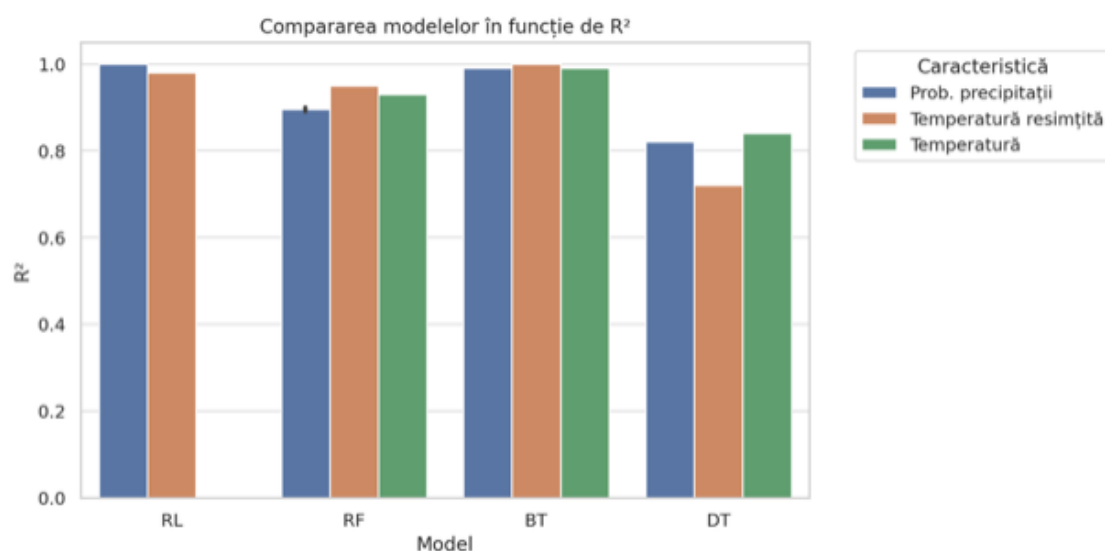
Figură 5.9 - Grafic cu compararea modelelor în funcție de MAE



Figură 5.10 - Grafic cu compararea modelelor în funcție de MSE



Figură 5.11 - Grafic cu compararea modelelor în funcție de RMSE



Figură 5.12 - Grafic cu compararea modelelor în funcție de R^2

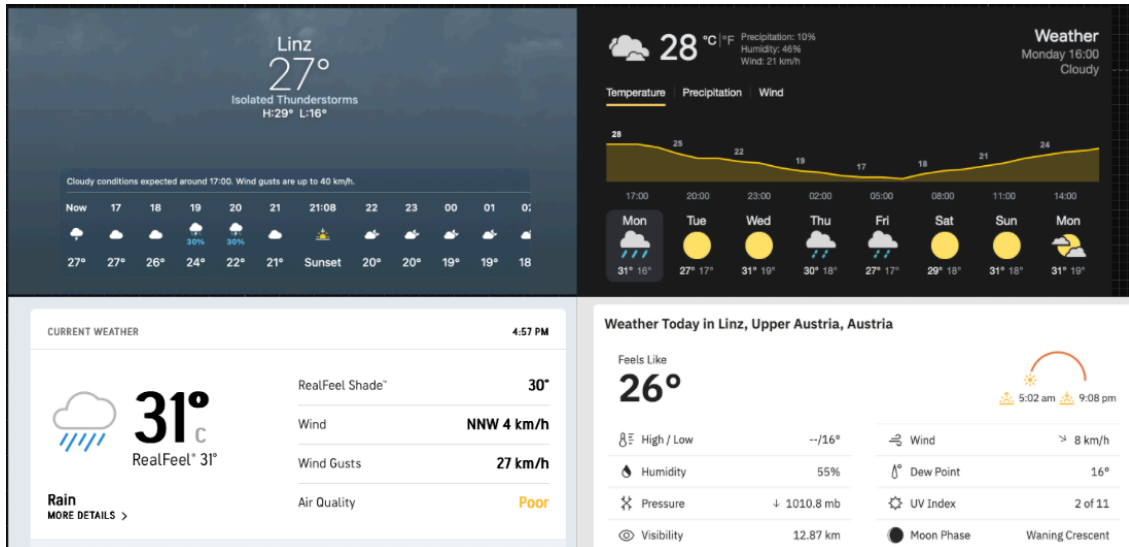
5.6. Comparație între SkyCast și alte aplicații meteorologice

Tabel 5.6 - Scorurile finale de evaluare folosind cei 4 regresori

| Regresor | Temperatura - °C | Temperatura resimțită - °C | Probabilitate de precipitații - % |
|-------------------|------------------|----------------------------|-----------------------------------|
| Linear Regression | 24.7 | 25 | 37 |
| Random Forest | 25.6 | 26.2 | 37 |
| Boosted Tree | 25.6 | 26.1 | 36 |
| Decision Tree | 24.5 | 24.6 | 21 |



Figură 5.13 - Rezultate meteorologie: OpenMeteo și WeatherAPI



Figură 5.14 - Rezultate meteorologice: Apple Weather, Google Weather, AccuWeather și The Weather Channel

Analizând rezultatele obținute de cei patru regresori utilizați în cadrul aplicației (Linear Regression, Random Forest, Boosted Tree, Decision Tree) și comparându-le cu valorile reale colectate în data de 23 iunie 2025, ora 18:00, din aplicații populare precum WeatherAPI, OpenMeteo, The Weather Channel, Google Weather, AccuWeather și Apple Weather, putem observa o serie de concluzii relevante privind acuratețea, robustețea și limitele fiecărui model.

Un prim aspect pozitiv este reprezentat de precizia estimărilor de temperatură ale modelelor folosind Linear Regression și Boosted Tee, care s-au apropiat foarte mult de intervalul raportat de majoritatea surselor oficiale. De exemplu, temperatura medie estimată de acești doi regresori a fost în jur de 24.7-26.6°C, valori care se aliniază destul de bine cu intervalul de 24-27°C raportat de Apple Weather, The Weather Channel și OpenMeteo. Acest lucru indică faptul că modelele implementate, chiar dacă sunt mai simple decât algoritmi avansați folosiți în industrie, pot totuși reproduce tendințe generale ale temperaturii în condiții obișnuite. De asemenea, și temperatura resimțită estimată de Linear Regression și Boosted Tree a fost apropiată de valorile raportate de aplicații, majoritatea fiind în jur de 25-26°C, semn că aceste modele sunt capabile să învețe relația complexă dintre temperatură, umiditate, vânt și alți factori care influențează senzația termică.

Pe de altă parte, un punct slab major este reprezentat de predicțiile referitoare la probabilitatea de precipitații. În special modelul Decision Tree a oferit o estimare semnificativ sub valoarea reală, indicând doar 21% șanse de ploaie într-o zi în care toate aplicațiile meteo importante raportau între 75% și 100% probabilitate. Deși modelele Random Forest și Boosted Tree au avut valori ceva mai ridicate (în jur de 36-37%), ele sunt totuși departe de realitatea atmosferică observată în Linz, conform surselor multiple. Această subestimare consistentă sugerează că modelele actuale nu sunt bine adaptate pentru a surprinde fenomenele de tip ploaie, furtuni sau alte condiții instabile. O posibilă explicație este că modelele au fost antrenate pe un set de date insuficient de divers și lipsit de variabile meteorologice cheie, cum ar fi presiunea atmosferică, viteza vântului sau date radar, care joacă un rol esențial în determinarea probabilității de precipitații.

O altă problemă identificată în urma acestei comparații este lipsa de consistență a estimărilor pe toate cele trei variabile (temperatură, temperatură resimțită și precipitații) în cazul modelului Decision Tree. Deși acest model este ușor de implementat și de interpretat, rezultatele sale au fost cele mai slabe în toate cazurile, atât din perspectiva erorilor (MAE,

RMSE, MSE), cât și în comparație cu valorile reale. În special, acest regresor pare să se suprapotrivească pe setul de date de antrenament, dar să generalizeze slab pe datele noi, reale, ceea ce este confirmat și de scorul R^2 mai scăzut față de ceilalți algoritmi. Pentru a remedia această problemă, o soluție ar fi fie ajustarea hiperparametrilor (cum ar fi micșorarea adâncimii maxime sau creșterea valorii minLossReduction), fie înlocuirea acestuia cu un model ensemble, precum Random Forest sau Gradient Boosted Tree, care se bazează tot pe arbori de decizie, dar reușesc să reducă erorile prin medierea predicțiilor din mai mulți arbori.

Totodată, rezultatele bune ale modelelor Boosted Tree și Random Forest în estimarea temperaturii sugerează că modelele ensemble sunt mai robuste și mai potrivite pentru sarcini de regresie în context meteorologic, în special atunci când datele de antrenament sunt relativ reduse. Boosted Tree a reușit să atingă un echilibru între bias și varianță, ceea ce explică performanțele mai bune în condiții complexe, în timp ce Random Forest a demonstrat o stabilitate crescută în predicție, reducând riscul de overfitting. Ambele ar putea beneficia în mod suplimentar de ajustarea fină a hiperparametrilor, dar și de creșterea volumului de date, atât pe orizont temporal, cât și pe dimensiunea variabilelor de intrare.

Pentru a îmbunătăți semnificativ acuratețea predicțiilor, este evident că trebuie extins setul de date prin integrarea unor parametri meteorologici suplimentari. Spre exemplu, includerea variabilelor precum presiunea atmosferică, nivelul de saturație a aerului, viteza rafalelor de vânt sau indicele UV ar putea permite modelelor să înțeleagă mai bine dinamica atmosferică, în special în contextul formării și anticipării precipitațiilor. În plus, ar fi oportun ca datele să fie colectate la o frecvență mai mare (de exemplu, la intervale de 15 minute în loc de orar) pentru a surprinde tranzițiile rapide între stările atmosferice.

Pe de altă parte, o posibilă direcție de îmbunătățire ar fi folosirea unor modele de tip rețea neuronală recurentă (RNN) sau LSTM (Long Short-Term Memory), care sunt concepute special pentru procesarea seriilor temporale și care pot învăța mai eficient patternuri sezoniere sau fluctuații rapide din atmosferă. În plus, integrarea unei componente de ensemble learning bazată pe medierea predicțiilor provenite din mai multe surse sau modele, inclusiv surse externe precum API-urile comerciale, ar putea duce la crearea unui sistem de predicție hibrid, capabil să combine avantajele modelelor interne cu precizia celor comerciale.

Așadar, analiza comparativă a valorilor prezise de regresorii implementați și a valorilor reale oferite de aplicațiile meteo confirmă faptul că, deși modelele simple oferă performanțe rezonabile în estimarea temperaturii și a temperaturii resimțite, acestea sunt încă departe de a fi suficiente pentru predicții meteorologice fiabile și de încredere, în special pentru variabile complexe precum probabilitatea de precipitații. Pentru a transforma aplicația într-un instrument cu utilitate practică reală, este esențială atât extinderea setului de date, cât și adoptarea unor modele algoritmice mai sofisticate, capabile să surprindă interacțiunile subtile dintre factorii meteorologici.

Capitolul 6. Concluzii

Proiectul prezentat urmărește să simplifice și să eficientizeze procesul de estimare a vremii pentru utilizatorii de aplicații mobile, prin implementarea unor algoritmi de regresie antrenați pe date meteo reale și prelucrați local. SkyCast, aplicația dezvoltată în acest scop, oferă utilizatorului posibilitatea de a consulta predicții pentru trei caracteristici esențiale: temperatură, temperatură resimțită și probabilitatea de precipitații, pe baza unor modele personalizate care se antrenează folosind atât date istorice, cât și surse de încredere precum OpenMeteo și WeatherAPI. Modelul se diferențează prin faptul că nu se bazează exclusiv pe estimările oferite de un API extern, ci încearcă să învețe din datele colectate în mod continuu pentru a furniza estimări locale personalizate.

Pentru dezvoltarea acestui sistem, au fost testați patru regresori diferiți: Linear Regression (LR), Random Forest (RF), Boosted Tree (BT) și Decision Tree (DT), fiecare evaluat în funcție de mai multe metrice precum MAE, RMSE, MSE și R^2 , dar și prin comparație directă cu valorile în funcție de aplicațiile comerciale de prognoză meteo. Un tabel comparativ a fost realizat pentru a evidenția performanțele obținute de fiecare model în cele trei variabile, iar rezultatele au indicat că BT și RF sunt cei mai potriviți pentru temperatură și temperatură resimțită, în timp ce toate modelele, inclusiv cei mai performanți, au întâmpinat dificultăți majore în predicția corectă a probabilității de precipitații.

Deși aplicațiile comerciale precum AccuWeather, The Weather Channel sau Apple Weather beneficiază de infrastructuri complexe, date radar și sisteme integrate de modelare atmosferică, rezultatele obținute de WeatherPrediction se apropie în mod surprinzător de estimările lor, cel puțin în ceea ce privește temperatura și temperatura resimțită. Totuși, diferențele majore observate la probabilitatea de precipitații evidențiază o limitare importantă a modelelor utilizate, în special din cauza absenței unor variabile atmosferice cheie și a lipsei unui set de date meteorologice mai cuprinzător.

Pe lângă funcționalitățile de bază care includ rularea modelelor și afișarea rezultatelor sub formă de listă sau grafic, aplicația integrează și posibilitatea de salvare a predicțiilor și a evaluărilor în fișiere CSV pentru analiză ulterioară. Această arhitectură modulară facilitează dezvoltarea viitoare și permite înlocuirea sau extinderea ușoară a modelului datorită separării clare între logica de prezentare și permite înlocuirea sau extinderea ușoară a modelului, datorită separării clare între logica de prezentare și cea de procesare a datelor. Modelul implementat, SkyCastModel, este proiectat astfel încât să accepte noi tipuri de regresori sau date suplimentare fără a necesita restructurări majore în aplicație.

Deși în forma sa actuală aplicația este funcțională și oferă rezultate promițătoare, există mai multe direcții de îmbunătățire ce pot fi luate în considerare. Una dintre acestea este integrarea unor modele specializate în procesarea seriilor temporale, cum ar fi LSTM sau alte rețele neuronale, care pot învăța mai eficient din evoluția condițiilor meteorologice pe termen scurt și mediu. De asemenea, extinderea setului de date de intrare cu variabile precum presiunea atmosferică, rafalele de vânt, nivelul de nebulozitate și indicii UV ar putea îmbunătăți semnificativ precizia, în special în cazul predicției de precipitații. Aceste date pot fi obținute fie prin extinderea surselor API existente, fie prin conectarea la baze de date meteorologice internaționale sau open-source.

Un alt aspect important este introducerea unei componente adaptive care să permită învățarea continuă din date înregistrate de utilizatori în timp real. Astfel, modelele ar putea deveni mai sensibile la schimbările climatice locale și ar putea învăța tipare specifice regiunii sau sezonului. De asemenea, o variantă interesantă ar fi testarea unor modele ensemble

hibride, care să combine estimările celor mai performanți regresori pe fiecare variabilă, crescând astfel acuratețea generală a aplicației.

Așadar, aplicația dezvoltată SkyCast demonstrează că este posibilă realizarea unei aplicații care să ofere predicții meteorologice rezonabil de precise folosind modele locale de machine learning, fără a depinde exclusiv de serviciile comerciale. Punctele forte ale aplicației sunt reprezentate de flexibilitatea în alegerea și antrenarea regresorilor, arhitectura modulară și scalabilitatea ridicată în timp ce limitele țin de complexitatea insuficient modelată a precipitațiilor și de lipsa unor date atmosferice avansate. Cu toate acestea, aplicația oferă o bază solidă pentru dezvoltări ulterioare și poate deveni, prin contribuții succesive, un instrument util în anticiparea vremii, mai ales în contexte personalizate sau locale.

6.1. Contribuții proprii

Contribuția principală a proiectului constă în dezvoltarea unei aplicații mobile de tip meteo, concepută pentru a realiza predicții locale prin utilizarea unor modele de regresie antrenate pe date reale, colectate și prelucrate constant. Diferența majoră față de alte aplicații existente pe piață este că WeatherPrediction nu se bazează exclusiv pe afișarea unor date preluate de la surse externe, ci permite rularea și evaluarea în timp real a patru regresori diferiți (Linear Regresion, Random Forest, Boosted Tree și Decision Tree), fiecare cu un set propriu de hiperparametri și performanțe măsurabile direct în aplicație. Experiența de utilizare este cât se poate de intuitivă și plăcută, ocupându-mă de întregul proces de dezvoltare al interfeței grafice și asigurând o structurare modulară a codoului, care permite extinderea ușoară cu noi funcționalități sau tipuri de regresori în viitor. Aplicația oferă flexibilitate în selectarea surse de date, a variabilei meteorologice de analizat și a momentului pentru care se face predicția, toate acestea fiind integrate într-un sistem coerent, ușor de folosit. Chiar dacă anumite aplicații comerciale includ funcții suplimentare precum afișarea grafică a radarului meteo sau predicții sezoniere, scopul acestui proiect a fost crearea unui instrument educațional și aplicativ care să permită testarea și compararea directă a performanței modelelor de regresie în predicția parametrilor atmosferici.

6.2. Dezvoltări ulterioare

Pentru a crește acuratețea predicțiilor și utilitatea practică a aplicației SkyCast, pot fi implementate o serie de îmbunătățiri și dezvoltări viitoare:

1. Utilizarea de algoritmi diferiți pentru fiecare parametru atmosferic

Această îmbunătățire ar putea aduce beneficii importante în ceea ce privește acuratețea predicțiilor. S-a observat că fiecare parametru are un comportament specific: temperatura are o variație mai lină și predictibilă, în timp ce probabilitatea de precipitații este mai stocastică și influențată de un număr mai mare de factori externi. Astfel, în loc să se utilizeze același model pentru toate predicțiile, se poate alege pentru fiecare variabilă un algoritm optim (de exemplu, Boosted Tree pentru temperatură și o rețea neuronală sau un model probabilistic pentru precipitații).

2. Extinderea setului de date și îmbunătățirea modului de validare

Ar reprezenta un pas esențial în rafinarea modelelor. În locul metodei implicite de validare automată, se propune trecerea la metode precum k-fold cross-validation, care asigură o estimare mai stabilă și mai obiectivă a performanței. De asemenea, antrenarea pe un volum mai mare de date (ex. cel puțin 30 de zile) și introducerea unei validări pe un set complet separat pot preveni overfitting-ul și pot simula mai bine situațiile reale.

3. Adăugarea de noi variabile meteorologice și creșterea frecvenței datelor

Poate contribui la o reprezentare mai fidelă a contextului atmosferic. Includerea unor factori suplimentari precum viteza și direcția vântului, presiunea atmosferică, indicele UV sau gradul de acoperire cu nori ar permite modelelor să surprindă mai bine relațiile cauzale care influențează temperatura resimțită sau probabilitatea de precipitații. De asemenea, colectarea și predicția la un interval de 15 minute, în loc de orar, ar spori sensibilitatea modelelor la schimbări rapide precum furtunile sau variațiile bruște de temperatură.

4. Integrarea unei funcționalități de locație dinamică și a unui sistem de comparare în timp real cu aplicațiile comerciale

Acest lucru ar îmbunătăți semnificativ experiența utilizatorului. Aplicația ar trebui să permită fie detectarea automată a locației prin GPS, fie introducerea manuală a unei locații dorite. În plus, introducerea unui modul comparativ care să prezinte, alături de predicțiile generate, și valorile curente oferite de aplicații consacrate precum AccuWeather, Apple Weather sau The Weather Channel, ar permite utilizatorului să înțeleagă diferențele și să decidă pe baza unei imagini de ansamblu.

5. Automatizarea selecției regresorului optim pentru fiecare situație

O asemenea îmbunătățire combinată cu posibilitatea învățării continue din datele noi colectate, ar transforma aplicația într-un sistem adaptiv. Aceasta ar permite alegerea automată a celui mai performant model în funcție de tipul variabilei, intervalul orar sau locația, iar modelele ar putea fi rafinate constant fără reantrenarea completă, prin metode precum incremental learning. În plus, salvarea predicțiilor și a valorilor reale ar facilita analiza retrospectivă și ar contribui la construirea unei baze solide pentru ajustări viitoare.

Bibliografie

- [1] OpenAI - <https://openai.com/index/chatgpt/>.
- [2] Open-meteo - https://open-meteo.com/en/docs?past_days=14&forecast_days=1&hourly=temperature_2m,apparent_temperature,precipitation_probability.
- [3] WeatherAPI - <https://www.weatherapi.com>
- [4] Grand View Research - <https://www.grandviewresearch.com/industry-analysis/weather-app-market-report>.
- [5] The Weather Channel - <https://weather.com/?Goto=Redirected>.
- [6] Weather Underground - <https://www.wunderground.com>.
- [7] Carrot Weather - <https://www.meetcarrot.com/weather/>.
- [8] Rasp Stephan. „WeatherBench: A Benchmark Data Set for Data-Driven Weather Forecasting", Journal of Advances in Modeling Earth Systems, 19 August 2020.
- [9] Weyn A. Jonathan. „Improving Data-Driven Global Weather Prediction Using Deep Convolutional Neural Networks on a Cubed Sphere", Journal Of Advances in Modeling Earth Systems, 12 August 2020.
- [10] Dark Sky - <https://darksky.org>.
- [11] ForecastWatch. „Most Accurate Weather App Identified (Soft-of)", 30 Ianuarie 2025.
- [12] Lerato.ro. „iOS sau Android? Pareri, avantaje și dezavantaje", 17 decembrie 2019.
- [13] Kecheng Qu. „Research on linear regression algorithm", Universitatea Shandong Xiehe, 2024
- [14] Wikipedia - https://en.wikipedia.org/wiki/Linear_regression.
- [15] Rajasekaran Meenal, Prawin Angel. „Weather prediction using random forest machine learning model", Research Gate, Mai 2021.
- [16] IBM. „What is random forest?" - <https://www.ibm.com/think/topics/random-forest>.
- [17] IBM. „What is a decision tree?" - <https://www.ibm.com/think/topics/decision-trees>.

Anexa 1. Exemple de cod

Citirea datelor externe

```

func fetchWeatherData() async {
    async let openModelCall = openNM.fetchWeatherData(latitude:
Constants.latitude,
                                longitude: Constants.longitude,
                                selectedDay: selectedDay)
    async let weatherModelCall = weatherNM.fetchWeatherData(latitude:
Constants.latitude,
                                longitude: Constants.longitude,
                                selectedDay: selectedDay)

    let (openModel, weatherModel) = await (openModelCall, weatherModelCall)

    guard let openModel = openModel as? OpenMeteoModel,
        let weatherModel = weatherModel as? WeatherAPIModel else { return }

    // print("OpenModel: \ \(openModel.hourly.time.count)")
    // print("WeatherAPI: \ \(weatherModel.forecast.forecastday.count *
weatherModel.forecast.forecastday.first!.hour.count)")
    // Uncomment this when we want the latest data in order to use it to train new
models

    Task { @MainActor [weak self] in
        guard let self else { return }
        buildCSVModel(openModel: openModel, weatherModel: weatherModel)
    }
}

```

Construirea modelului CSV

```

private func buildCSVModel(openModel: OpenMeteoModel, weatherModel:
WeatherAPIModel) {
    for i in 0..<openModel.hourly.time.count {
        let currentTime = openModel.hourly.time[i]
        mlModel.append(SkyCastModel(latitude: Constants.latitude,
                                longitude: Constants.longitude,
                                time: currentTime,
                                omTemp: openModel.hourly.temp[i],
                                omFeelLike: openModel.hourly.feelLikeTemp[i],
                                omPrecipProb: openModel.hourly.precipProb[i]))
    }

    for forecastDay in weatherModel.forecast.forecastday {
        for hour in forecastDay.hour {
            if let index = mlModel.firstIndex(where: { $0.time == hour.time }) {
                mlModel[index].wTemp = hour.temp
                mlModel[index].wFeelLike = hour.feelLikeTemp
                mlModel[index].wPrecipProb = hour.precipProb
            }
        }
    }
}

```

```
    }  
  }  
}  
  
mlModel.sort(by: {  
  guard let t1 = $0.time, let t2 = $1.time else { return false }  
  return t1 < t2  
})  
  
exportToCSV()  
}
```

Calcularea metricilor de evaluare

```
func computeAccuracyMetrics(actualValues: [Double], predictedValues: [Double],  
target: String) {  
  let n = Double(actualValues.count)  
  
  let mae = zip(actualValues, predictedValues).map { abs($0 - $1) }.reduce(0, +) /  
n  
  let mse = zip(actualValues, predictedValues).map { pow($0 - $1, 2) }.reduce(0,  
+) / n  
  let rmse = sqrt(mse)  
  
  let meanActual = actualValues.reduce(0, +) / n  
  let ssTotal = actualValues.map { pow($0 - meanActual, 2) }.reduce(0, +)  
  let ssResidual = zip(actualValues, predictedValues).map { pow($0 - $1, 2)  
}.reduce(0, +)  
  let r2 = 1 - (ssResidual / ssTotal)  
  let evMetric = EvaluationMetric(target: target, mae: mae, mse: mse, rmse: rmse,  
r2: r2)  
  evaluationMetrics.append(evMetric)  
  predictedCSVString.append("\(evMetric.description)\n")  
}
```

Anexa 2 Lista figurilor și a tabelelor

| | |
|--|----|
| Figură 4.1 - Diagramă de flow | 11 |
| Figură 4.2 - Regresia liniară [14] | 15 |
| Figură 4.3 - Random Forest [16] | 16 |
| Figură 4.4 - Regresie cu arbori de decizie [17] | 17 |
| Figură 4.5 - Structura fișierelor | 19 |
| Figură 4.6 - Interfața grafică în modul grafic | 20 |
| Figură 4.7 - Interfața grafică în modul listă | 21 |
| Figură 4.8 - Structura modelului de date OpenMeteoModel | 22 |
| Figură 4.9 - Structura modelului de date WeatherAPIModel | 23 |
| Figură 4.10 - Structura modelului de date SkyCastModel | 24 |
| Figură 4.11 - Pregătirea modelului folosind Create ML | 25 |
| Figură 4.12 - Evaluarea modelului folosind Create ML | 26 |
| Figură 4.13 - Detalii despre model în Create ML | 27 |
| Figură 4.14 - Crearea modelului în mod programatic | 28 |
| Figură 4.15 - Structura fișierului CSV cu date de intrare | 30 |
| Figură 4.16 - Conținutul fișierului Predictions.csv | 31 |
| Figură 5.1 - Valorile metricilor de evaluare folosind regresie liniară cu valori implicite pentru hiperparametri | 32 |
| Figură 5.2 - Evoluția MAE în funcție de numărul de iterații folosind Linear Regression | 35 |
| Figură 5.3 - Evoluția R^2 în funcție de numărul de iterații folosind Linear Regression | 35 |
| Figură 5.4 - Evoluția MSE în funcție de numărul de iterații folosind Linear Regression | 36 |
| Figură 5.5 - Evoluția RMSE în funcție de numărul de iterații folosind Linear Regression | 36 |
| Figură 5.6 - Valorile metricilor de evaluare folosind Random Forest cu valori implicite pentru hiperparametri | 37 |
| Figură 5.7 - Valorile metricilor de evaluare folosind Boosting Tree cu valori implicite pentru hiperparametri | 39 |
| Figură 5.8 - Valorile metricilor de evaluare folosind Decision Tree cu valori implicite pentru hiperparametri | 42 |
| Figură 5.9 - Grafic cu compararea modelelor în funcție de MAE | 46 |
| Figură 5.10 - Grafic cu compararea modelelor în funcție de MSE | 46 |
| Figură 5.11 - Grafic cu compararea modelelor în funcție de RMSE | 47 |
| Figură 5.12 - Grafic cu compararea modelelor în funcție de R^2 | 47 |
| Figură 5.13 - Rezultate meteorologie: OpenMeteo și WeatherAPI | 47 |
| Figură 5.14 - Rezultate meteorologie: Apple Weather, Google Weather, AccuWeather și The Weather Channel | 48 |
| Tabel 3.1 - Comparatie Apple Weather, Google Weather și The Weather Channel | 8 |
| Tabel 5.1 - Evaluarea performanței folosind regresie liniară | 33 |
| Tabel 5.2 - Evaluarea performanței folosind Random Forest | 38 |
| Tabel 5.3 - Evaluarea performanței folosind Boosted Tree | 40 |
| Tabel 5.4 - Evaluarea performanței folosind Decision Tree | 43 |
| Tabel 5.5 - Cele mai bune rezultate ale regresorilor folosiți în evaluarea modelului | 44 |

| | |
|---|----|
| Tabel 5.6 - Scorurile finale de evaluare folosind cei 4 regresori | 47 |
|---|----|