UNIVERSITATEA DE VEST DIN TIMIŞOARA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
PROGRAMUL DE STUDII DE MASTERAT:
Bioinformatică

LUCRARE DE DISERTAȚIE

COORDONATOR:
Conf. Dr. Onchiș Darian

ABSOLVENT:
Babuc Diogen

TIMIŞOARA
2023

UNIVERSITATEA DE VEST DIN TIMIŞOARA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
PROGRAMUL DE STUDII DE MASTERAT:
Bioinformatică

Modele predictive de învățare automată aplicate seturilor de date COVID-19 pentru estimarea riscului de infecție în anumite regiuni geografice

COORDONATOR:
Conf. Dr. Onchiș Darian

ABSOLVENT:
Babuc Diogen

TIMIŞOARA
2023

Rezumat

Acest studiu investighează indicatorii statistici și metricile de evaluare a performanței pentru modelele de învățare automată, aplicate pe două seturi de date COVID-19. Scopul este găsirea celui mai performant model care urmărește progresul statisticilor COVID-19. Modelele selectate sunt K-Nearest Neighbors, Decision Tree (DT), Support Vector Machine, Classification and Regression Trees (CART) și Extreme Gradient Boost (XGBoost). Setul de date pentru țările din Oceania a avut 10 observații, însă a fost mărit la 110 pentru a avea suficiente date de antrenare. Cele mai performante modele pentru acest set de date sunt CART (DT cu criterion='gini') și DT (criterion='entropy'), cu acuratețea, sensibilitatea, specificitatea, precizia și scorul F1 peste 93.2%. Valorile lui DT pentru eroarea medie pătratică sunt între 0.3 și 3.5, în timp ce pentru eroarea medie absolută sunt mai mici de 1.7. Coeficientul de determinare este, în medie, 0.8 atât pentru CART, cât și pentru DT. Coeficientul de corelație nu este mai mic de 0.81. Pentru setul de date cu ultimele 100 de țări de pe Worldometer (cazuri totale până în septembrie 2020), modelele CART și DT au o acuratețe și specificitate de peste 88%; precizia și scorul F1 sunt peste 78%; sensibilitatea pentru DT este de aproximativ 84%. Graficele și diagramele, obținute pentru modelul DT, arată că eroarea medie pătratică și cea absolută au o valoare mai mică de 2. XGBoost se află pe a treia poziție. Unele modele pot ajuta la descoperirea regiunilor geografice cu risc de infecție. CART și DT sunt modelele cele mai performante pentru ambele seturi de date selectate.

Cuvinte cheie: COVID-19, Modele de învățare automată, Evaluarea modelelor, Seturi de date Kaggle, Statistică

Abstract

This study investigates statistical indicators and evaluation metrics for machine learning models on two COVID-19 datasets. The goal is to find the best-performing model which tracks the progress of COVID-19 statistics. The selected models are K-Nearest Neighbors, Decision Tree (DT), Support Vector Machine, Classification and Regression Trees (CART), and Extreme Gradient Boost (XGBoost). The dataset for the Oceania countries, selected from Kaggle, had 10 observations but was augmented to 110, using the cumulative sum of the values from two columns, to have enough training data. The best-performing models for this dataset are CART (DT with criterion='gini') and DT (criterion='entropy'), with accuracy, sensitivity, specificity, precision, and accuracy above 93.2%. The DT values for mean squared error are between 0.3 and 3.5, while for mean absolute error are less than 1.7. The coefficient of determination is, on average, 0.8 for both CART and DT. The correlation coefficient is not less than 0.81 for DT, and not less than 0.83 for CART. For the dataset with the last 100 countries on Worldometer (total cases up to September 2020), the CART and DT models achieve accuracy and specificity above 88%; precision and accuracy are above 78%; sensitivity for DT is around 84%. The graphs and charts, obtained for the DT model, show that mean squared and absolute errors have a value of less than 2. XGBoost is in the third position. Some models can help in discovering geographic regions with a risk of infection. CART and DT are the best-performing models for both selected datasets.

Keywords: COVID-19, Machine Learning, Models Evaluation, Kaggle Datasets, Statistics

Cuprins

Capitolul 1

Introducere

Acest capitol examinează motivația practică a acestui studiu. Se analizează importanța științifică a tematicii și se indică obiectivele principale ale studiului. Se efectuează o intrare simplă în tematica selectată și în conținut.

1.1 Context

Problema principală a acestei lucrări este calculul complexității modelelor în detectarea zonelor cu risc de infectare cu SARS-CoV-2, care provoacă boala COVID-19. Investigăm performanța modelelor de învățare automată pentru anumite regiuni geografice (țări/teritorii din Oceania și ultimele 100 de țări în funcție de numărul total de cazuri de COVID-19 din septembrie 2020). Modelele de învățare automată sunt comparate între ele folosind diferiți parametri ai metodelor. Modelele selectate sunt K-Nearest Neighbors, Decision Tree (criterion='entropy'), Support Vector Model, Clasification and Regression Trees (Decision Tree cu criterion='gini') și Extreme Gradient Boost. Investigăm indicatorii statistici și metricile de evaluare a performanței modelelor. Având în vedere cantitățile mari de date, generate de studiile COVID-19, modelele de învățare automată oferă o modalitate puternică de a analiza și înțelege aceste date. Teoreticienii pot dezvolta o înțelegere mai profundă a virusului. Putem lucra împreună pentru a găsi o soluție la această criză globală de sănătate. În acest context, această lucrare explorează diferitele modele de învățare automată aplicate setului de date COVID-19; includem punctele forte, limitările și potențialele implicații în cercetarea și politica domeniul sănătății publice. A apărut întrebarea cu ce metode de clasificare și regresie să se ocupe. Conform literaturii de specialitate, indicatorii statistici aleși sunt eroarea medie pătratică, eroarea medie absolută, coeficientul de determinare și coeficientul de corelație. Un alt indicator statistic, selectat pentru a soluționa modelele cu niște valori implicite, este aria sub curba ROC. Măsurile de evaluare a performanței utilizate în studiu sunt acuratețea, sensibilitatea, specificitatea, precizia și acuratețea. Modelul de regresie liniară este utilizat pentru a face o predicție; construim variabila independentă și cea dependentă și antrenăm modelul. Programul necesită date de intrare și obține predicția asupra numărului total de cazuri de infecție pentru o țară. Revizuirea literaturii de specialitate este semnificativă în alegerea metodelor de examinare a datelor și în alegerea modelelor de învățare automată.

1.2 Scop și obiective

Secțiunea care indică scopul și obiectivele studiului pune bazele unei cercetări științifice. Această secțiune oferă o declarație clară și concisă despre ce se discută în cercetare și despre ce își propune autorul să realizeze.

Scopul acestui experiment este urmărirea progresului de infecție și diminuarea efectului de epidemie în anumite regiuni geografice: ultimele 100 țări de pe Worldometer (cazuri totale până pe septembrie 2020) și Oceania, folosind cele mai performante modele de învățare automată.

Obiectivele conturează scopul cercetării și oferă o direcție clară pentru studiu. Obiectivele sunt specifice, realizabile și relevante. Cele patru obiective principale ale acestui studiu sunt:

- dezvoltarea unui sistem de procesare a datelor,
- evaluarea modelelor de învățare automată pentru rezultatele COVID-19 folosind o varietate de indicatori și metrici statistici,
- compararea modelelor de învățare automată dezvoltate și
- construirea unui model de predicție (model de regresie liniară).

Primul obiectiv implică curatarea, transformarea și procesarea seturilor de date COVID-19 pentru a fi utilizate în modele de învățare automată. Aceasta implică identificarea valorilor lipsă, a valorilor aberante și a altor anomalii. Al doilea obiectiv implică dezvoltarea, instruirea și testarea mai multor modele de învățare automată și evaluarea performanței utilizând metricile: acuratețe, precizie, acuratețe, sensibilitate și specificitate. Indicatorii statistici au scopul de a justifica selecția și de a regla modelele de învățare automată. Al treilea obiectiv analizează modelele predictive pentru rezultatele COVID-19 și identifică cele mai performante modele pentru problema studiului de față. Diagramele construite diferențiază modelele în funcție de performanță. Al patrulea obiectiv estimează valoarea concretă a numărului total de persoane infectate dintr-o țară/teritoriu, la o perioadă îndepărtată.

Precizarea scopului și a obiectivelor lucrării ajută la stabilirea semnificației cercetării, a fezabilității și a impactului potențial.

1.3 Structura lucrării de disertație

În această secțiune se va povesti pe scurt despre fiecare capitol, secțiune și subsecțiune.

În Capitolul 2 se discută despre conceptele teoretice ale acestui studiu. Se definesc noțiunile teoretice fundamentale: COVID-19 și modelele utilizate și li se indică rostul. În Secțiunea 2.1, unde se discută despre COVID-19, s-a indicat și un exemplu concret, din sfera medicinei, în care se specifică modul de infecție cu virusul *SARS-CoV-2*. În Secțiunea 2.2 se elaborează abordările deja existente; se trage o paralelă între literaturile de specialitate cu privire la modelele de învățare automată, indicatorii statistici și metricile de evaluare a performanței utilizate. S-a analizat și modul în care au fost abordate datele de preprocesare. În Secțiunea 2.3 s-au precizat modelele utilizate în această lucrare. În Secțiunea 2.4 s-au discutat metodele pentru evaluarea modelelor cu privire la statistica de descriere, indicatorii statistici și metricile de evaluare.

În Capitolul 3 se discută despre modul în care este organizată aplicație și despre modul în care se utilizează modelele selectate. Se menționează rostul modelelor în lumea reală. Se precizează limitările modelelor actuale și cum pot fi îmbunătățite. Se discută implicațiile etice ale utilizării modelelor de învățare automată în domeniul sănătății. Se stabilesc potențialele preocupări legate de confidențialitate și securitatea datelor. Implicații potențiale de politică pentru utilizarea modelelor de învățare automată în domeniul sănătății. Puteți discuta potențiale cadre de reglementare sau politici care ar putea fi puse în aplicare pentru a asigura utilizarea responsabilă a acestor modele.

În Secțiunea 3.1 se descrie modul de selecție a seturilor de date. Seturile de date sunt preluate de pe platforma Kaggle. Sunt selectate doar anumite observații din seturile de date de pe Kaggle. Ambele seturi de date au fost augmentate de 10 ori, cu rostul de a ajunge la un volum dorit, pentru ca modelele să funcționeze cu o performanță ridicată. În Secțiunea 3.2 este explicată partea de preprocesare și de prelucrare a datelor. Setul de date CSV este convertit într-un cadru de date (data frame), iar valorile aberante și cele lipsă sunt înlăturate și completate cu media valorilor de pe acea coloană. În Secțiunea 3.3 sunt indicate modalitățile de construire a indicatorilor statistici, specificându-se metodele și funcțiile necesare din anumite module. În Secțiunea 3.4 se construiesc metricile de evaluarea a performanței modelelor. Sunt indicate clasele și funcțiile, precum și modulele și librăriile folosite pentru proiectare și reprezentarea grafică. În Secțiunea 3.5 se creează diagrama de sistem, în Visual Paradigm, în procesul de construire a modelelor de învățare automată. În Secțiunea 3.6 este explicat procedeul de implementare a experimentului în cadrul de lucru Streamlit, pentru a obține o interfață/dashboard cu prelucrările realizate și rezultatele obținute. Acest cod este schimbat pentru a fi compatibil cu platforma Streamlit Cloud. Sunt precizate instrucțiunile aferente urcării aplicației pe acest cloud, precum și fișierele adițional create, necesare pentru a face aplicația disponibilă tuturor. În Secțiunea 3.7 este specificat fluxul aplicației. Acesta presupune precizarea paginilor accesate și a conținutului acestora. În Secțiunea 3.8 se găsește specificația de implementare, prin indicarea celor mai importante secvențe de cod și explicația acestora.

În Capitolul 4 sunt reprezentate rezultatele aferente indicatorilor statistici și ale modelelor de învățare automată, pentru ambele seturi de date, urmând două criterii de evaluare a modelelor. Se construiește câte un grafic pentru fiecare model, pentru a observa eroarea medie pătratică, eroarea medie absolută, coeficientul de determinare și cel de corelație, un parametru selectat pentru fiecare model variază de la 1 la 40. Metricile de evaluare a performanței sunt puse într-o singură figură. Sunt furnizate rezultatele pentru acuratețe, sensibilitate, specificitate, precizie și scorul F1. Este precizat cel mai performant model dintre cele alese pentru experiment. În Secțiunea 4.1 sunt menționate valorile indicatorilor statistici. În Subsecțiunea 4.1.1. este realizată clasificarea setului de date despre țările din Oceania, privind raportul dintre două coloane, cazurile totale și populație. În Subsecțiunea 4.1.2, clasificarea se săvârșește numai după *Total Cases*. Se estimează țările cu risc de infecție, și cele fără risc, pe datele de antrenare din setul de date despre țările din Oceania. În Secțiunea 4.2 se calculează erorile medii, coeficientul de determinare și corelație, și aria sub curba ROC pentru modelele de învățare automată privind setul de date cu ultimele 100 țări de pe Worldometer (cazuri totale de infectați până pe septembrie 2020). În Subsecțiunea 4.2.1 este indicată etapa de clasificare, cu rostul de a obține valorile pentru metricile de evaluare a performanței. În Subsecțiunea 4.2.2, clasificarea se realizează folosind

numai o limită pentru coloana de cazuri totale de infectați, urmând câmpurile de valori din setul de date cu ultimele 100 țări de pe Wordometer. Secțiunea 4.3 menționează pașii spre crearea unui model liniar de predicție (folosind *LinearRegression*), pentru a obține o valoare de cazuri totale în conformitate cu valorile de pe Worldometer, cu coloanele pentru variabila independentă.

În capitolul 5 se indică rezultatele cele mai semnificative pentru fiecare set de date, se specifică limitările sistemului studiului și se propun soluții potențiale în rezolvarea limitărilor. Se face o recapitulare a întregului proiect, specificând avantajele și dezavantajele studiului și a sistemului dezvoltat, precum și soluțiile potențiale pentru problemele care au apărut. Bioinformaticienii intrigați în această tematică pot prelua ideea, cu scopul de a o dezvolta și pentru a propune optimizări utile.

Capitolul 2

Starea de artă

Acest capitol este o parte importantă a oricărui studiu deoarece oferă o bază pentru cercetarea care urmează. Capitolul respectiv oferă informații despre subiectul studiat, inclusiv teoriile și conceptele găsite în studiu. Aceste informații pot ajuta cititorii să înțeleagă contextul cercetării și importanța acesteia.

2.1 COVID-19

COVID-19 este o boală respiratorie infecțioasă cauzată de virusul SARS-CoV-2. A fost identificată pentru prima dată în Wuhan, China, în decembrie 2019, și de atunci s-a răspândit rapid pe tot globul, ducând la o pandemie. Virusul se răspândește prin picături respiratorii, atunci când o persoană infectată respiră aproape de cealaltă, tușește sau strănută. Simptomele COVID-19 pot varia de la ușoare la severe, și pot include febră, tuse, oboseală, pierderea gustului sau a mirosului, și altele. Unele persoane pot fi asimptomatice, în timp ce altele pot necesita terapie intensivă. Măsurile preventive împotriva COVID-19 includ purtarea măștilor, distanțarea socială, spălarea frecventă a mâinilor, evitarea adunărilor mari și vaccinarea. Vaccinurile au fost dezvoltate și sunt distribuite în întreaga lume pentru a combate răspândirea virusului [1]. Discuția despre COVID-19 este una de actualitate și, fiind student la Bioinformatică, autorul acestei lucrări a ales să analizeze două seturi de date corelate cu această boală. Se folosește o mulțime de parametri și procedee pentru dezvoltarea tematicii, pentru estimarea zonelor geografice cu risc de infecție peste medie, folosind modele de învățare automată. Aceste modele sunt implicate în diferite procese privind subdomeniul medicinei care se ocupă cu studiul virușilor, în special cu cel mai popular virus din ultimii trei ani și jumătate, SARS-CoV-2. Algoritmii de învățare automată se pot încadra în mulțimea abordărilor utilizate pentru detecția unui pacient infectat cu acest virus. Pandemia de COVID-19 a provocat o criză globală de sănătate, care a afectat și continuă să afecteze milioane de oameni din întreaga lume. Cercetătorii și oamenii de știință din sfera de date au apelat la modelele de învățare automată pentru a ajunge la o mai bună înțelegere a virusului și a impactului acestuia. Modelele de învățare automată s-au dovedit a fi un instrument valoros pentru analiza seturilor de date COVID-19, permițând cercetătorilor să identifice pericolele și efectele infecției, să facă predicții și să dezvolte noi tratamente, printre care se evidențiază vaccinurile.

Exemplu concret

Un fapt destul de interesant, care merită să fie menționat în cadrul acestei secțiuni textuale, este o secvență completă de genom, care a fost obținută pentru o tulpină de coronavirus, cu sindrom respirator acut sever (SARS). Această tulpină a fost extrasă prin calea orofaringiană a unui pacient nepalez, care s-a întors în Nepal, din Wuhan, China. Acest individ este un student la Universitatea de Tehnologie din Wuhan, China, și are 32 de ani. Imaginea sa medicală este una curată, fără condiții concomitente [2]. Acest virus, cu sindromul respirator acut sever este din familia *Coronaviridae*, genul *Betacoronavirus*. După cum este cunoscut acum, virusul s-a răspândit pe scară largă în China. A afectat și alte țări, cum sunt Hong Kong, Singapore, Nepal și Japonia.

În cadrul articolului respectiv, care reprezintă un studiu de caz concret în procedeul de investigație al infectării cu acest virus, se raportează secvența completă a genomului de la un pacient nepalez; infecția a fost dobândită în Wuhan, China și importată în Nepal. Izolatul respectiv provine din specimenul de tampon orofaringian al unui bărbat care a depășit cu puțin a treia decadă, totodată și student nepalez în Wuhan, fără antecedente de condiții concomitente. Simptoamele pe care le-a avut studentul atunci când s-a întors în Nepal sunt febră ușoară, durere în gât și tuse, ceea ce, ulterior s-a dovedit, sugerează boala COVID-19. Laboratorul Național de Sănătate Publică din Kathmandu, Nepal, a colectat eșantionul de tampon orofaringian, care a fost transmis la un laborator de la Universitatea din Hong Kong, unde a fost și confirmat cazul, folosind procedeul de secvențiere. Exemplul concret, specimenul, a fost testat pozitiv pentru virusul *SARS-CoV-2* folosind abordarea de PCR în timp real, dezvoltată la Universitatea din Hong Kong. Procesul de secvențiere a fost realizat folosind sistemul *Illumina MiSeq* și algoritmul de asamblare *Burrows-Wheeler Aligner MEM*, abreviat BWA-MEM. Din extractul de ARN a fost amplificat genomul, ținându-se cont de specimenul original. Noua secvență a fost obținută prin cartografierea inițialelor citiri la un genom de referință *SARS-CoV-2*. Și la acest pas s-a folosit algoritmul BWA-MEM cu atribútele și caracteristicile implicite, pentru a determina secvența consens. Adițional la aceasta, ansamblul produs de *Megahit*, cunoscut ca și ansamblu de novo, a fost folosit pentru metoda de validare încrucișată, cu metoda bazată pe referințe de control intern. Cele două rezultate au fost consistente, iar secvența finală s-a sprijinit pe metoda bazată pe referințe. Secvența de referință utilizată în acest sector de analiză este obținută de la *Global Initiative on Sharing All Influenza Database* [3]. Ulterior, citirile de referință s-au aliniat folosind un pile-up, cu țelul de a se ajunge la pragul minim de acoperire. De asemenea, cu rostul de a evalua calitatea unei secvențe înaintea procesului de tăiere, și după efectuarea alinierii de prevenire a unei greșeli, a fost utilă o tehnologie cunoscută sub denumirea de *FastQC*. Este, în esență, un fișier cu foarte multe secvențe; pe exemplul de față chiar aproape de zece milioane de observații, printre care se enumeră și peste cinci milioane de secvențe cu o pereche terminală.

S-a generat o secvență de aproape 30 perechi de bază. Situsurile de la capetele 5' și 3' au fost îndepărtate, de unde se va putea constata că genomul cu circa 60 nucleotide este mai scurt decât genomul de referință din GenBank, al cărui număr de acces este NC_045512. Folosind algoritmul *Clustal W* s-au aliniat secvențele genomului complet. [4] Noua secvență a virusului a fost comparată cu câțiva genomi deja existenți în *NCBI BLAST*. După ce s-a comparat genomul izolatului cu încă doi genomi, *MN988668* [15] și *NC_045512* [16], secvențiați și disponibili în GenBank pentru localitatea Wuhan, s-a găsit o potrivire aproape complet exactă între acestea. S-a constatat și o asemănare

de peste 99.9% a genomului izolatului cu încă șapte secvențe, în total [5].

2.2 Modele de învățare automată

Învățarea automată este o sub-ramură a inteligenței artificiale care implică antrenarea de modelor, urmând datele dintr-un set de date, pentru a recunoaște tipare, a face predicții, a clasifica datele, și multe altele. În cazul COVID-19, modelele de învățare automată sunt folosite pentru a analiza datele concomitente cu rata de infecție, sau alte valori similare. În studiul de față, modelele monitorizează răspândirea virusului; se identifica populațiile cu risc ridicat. Rezultatele obținute pot ajuta cercetătorii specialiști la dezvoltarea noilor tratamente. Modelele de învățare automată devin din ce în ce mai importante pe măsură ce pandemia continuă să evolueze. Este un instrument din ce în ce mai utilizat pentru analiza seturilor de date complexe, inclusiv cele legate de sănătatea publică. În contextul pandemiei COVID-19, cercetătorii au folosit algoritmi din acest domeniu pentru a dezvolta modele predictive care pot ajuta la identificarea efectului infecției cu virusul SARS-CoV-2 și la potențialii factori de risc asociați cu acest virus. Unele dintre cele mai frecvent utilizate modele în acestă problematică sunt K-cei mai apropiați vecini, Modelul cu suport vectorial, Arborele de decizie (cu parametrul `criterion='gini'`), Extreme Gradient Boost, și Arborii de clasificare și regresie (Arbore de decizie cu parametrul `criterion='gini'`).

K-cei mai apropiați vecini (KNN) este un algoritm simplu și puternic care se bazează pe metrici de distanță pentru a clasifica punctele de date. Este adesea folosit în diagnosticul medical și studiile epidemiologice, inclusiv COVID-19. Modelul cu suport vectorial este un algoritm mai complex decât KNN, și este potrivit pentru sarcinile de clasificare care implică seturi mari de date, cu o dimensionalitate ridicată. Arborele de decizie este o alegere utilă pentru construirea modelelor de luare a deciziilor deoarece este ușor de analizat și poate cuprinde relații complexe între variabile. Extreme Gradient Boost este un model puternic, mai nou decât modelele anterioare. Este un algoritm de învățare în ansamblu, care combină mai mulți arbori de decizie pentru a îmbunătăți acuratețea și generalizarea datelor. În cele din urmă, CART este un algoritm similar cu Arborele de decizie. Utilizează împărțiri de tip binar cu scopul de a construi un arbore de decizie. Aceste modele de învățare automată sunt utilizate pentru a dezvolta modele predictive pentru o varietate de rezultate legate de COVID-19, inclusiv progresul bolii, infecția și rata de transmitere. Analizând seturi mari de date ale pacienților cu COVID-19, aceste modele pot identifica factori de risc importanți și pot ajuta furnizorii de asistență medicală și factorii de decizie la diferite intervenții de sănătate publică.

2.3 Abordări existente

Această secțiune are o semnificație importantă în alegerea, crearea și dezvoltarea parametrilor pentru analiza comparației dintre modelele de învățare automată. Sunt listate articolele folosite. Ele servesc ca suport, bază și motivație pentru ideea de a proiecta un proiect cu mai multe straturi în domeniile învățării automate, învățării profunde, oncologiei și statisticii virale. Rezultatele sunt concrete și obținute prin indicatori statistici și metrici de evaluare. Articolele citite și studiate oferă procente pentru acurateță, sensibilitate, specificitate, precizie și scorul F1, pentru partea de instruire, validare și

testare. Validarea și testarea sunt uneori tratate împreună. Unele articole indică faptul că este util să se calculeze performanța modelului prin intermediul unor metrici de evaluare, cum ar fi sensibilitatea și specificitatea [1]. Alte articole precizează, de asemenea, că acuratețea, scorul F1 și precizia sunt indicatori utili [2]. Autorii articolului *COVID-19 Diagnosis Prediction in Emergency Care Patients: A Machine Learning Approach* au aplicat cinci metode de învățare automată: Neural Networks, Random Forest (RF), Extreme Gradient Boost (XGBoost), Logistic Regression (LR) și Modelul cu suport vectorial (SVM), pentru a investiga pericolul COVID-19 cu un diagnostic pozitiv și a luat în considerare cincisprezece caracteristici predictive. Datele au fost colectate de la două sute de pacienți adulți din spitalul israelit numit *Albert Einstein*, din Brazilia [3]. Performanța predictivă a fost măsurată prin calcularea ariei de sub curba ROC (Receiver Operating Characteristics) (AUC), a specificității, sensibilității, scorului Brier, scorului F1, a valorilor de predicție pozitive și negative.

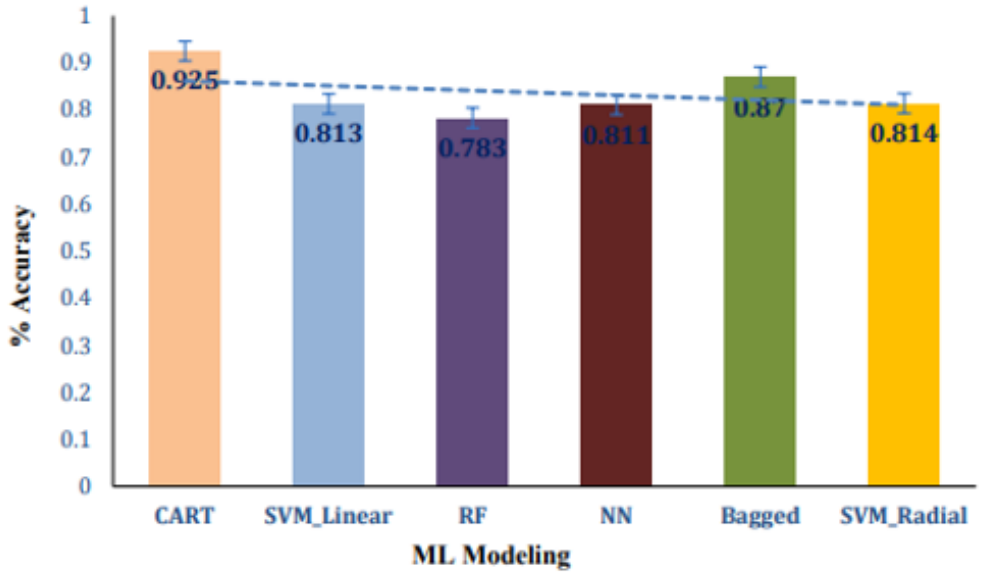

Figura 2.1: Acuratețea modelelor analizate urmând aria de sub curbă [6].

Cea mai bună performanță predictivă a fost obținută de algoritmii SVM și RF cu un AUC de 0.85 pentru setul de testare. Modelul XGBoost a fost folosit pentru a prezice rezultatele COVID-19 și a fost cel mai performant. Modelul RF a fost folosit pentru a prezice internările în spital pentru acei pacienți care sunt pozitivi la SARS-CoV-2. Cea mai benefică metodă este Arborele de clasificare și regresie (CART), cu o acuratețe de 92.5%.

Un alt studiu, *Machine Learning Applied to the Diagnosis of Human Diseases: A Systematic Review* a inclus cinci abordări care sunt binecunoscute în domeniul învățării automate. Acele abordări sunt Decision Tree (DT), SVM, Artificial Neural Network, CART și RF, în timp ce XGBoost a fost luat în considerare în procesul de clasificare, pentru care autorii au calculat sensibilitatea și specificitatea pentru a obține o măsură de performanță. Un alt studiu, *Towards an Artificial Intelligence Framework*

for Data-Driven Prediction of Coronavirus Clinical Severity, a prezis severitatea clinică a coronavirusului folosind tehnici de învățare automată. Setul de date a fost obținut de la două spitale din Wenzhou, China. Autorii au luat în considerare unsprezece caracteristici predictive și au aplicat diferiți clasificatori, cum ar fi LR, K-Nearest Neighbors, DT, SVM și RF [5]. Clasificatoarele au fost testate luând în considerare doar valorile de precizie. SVM a arătat cea mai bună acuratețe, cu o valoare de 80%.

Într-un alt studiu [9], autorii au evaluat o matrice de corelație care oferă un diagnostic al conexiunii pereche dintre fiecare variabilă, folosind coeficientul de corelație Pearson. Acest studiu s-a concentrat pe analiza corelației folosind eticheta clasei. Unele caracteristici au arătat relații puternice. Matricea de corelație arată o rată pozitivă ridicată, apropiată de 1.0, pentru variabilele *hematocrit*, RDW și *hemoglobină*.

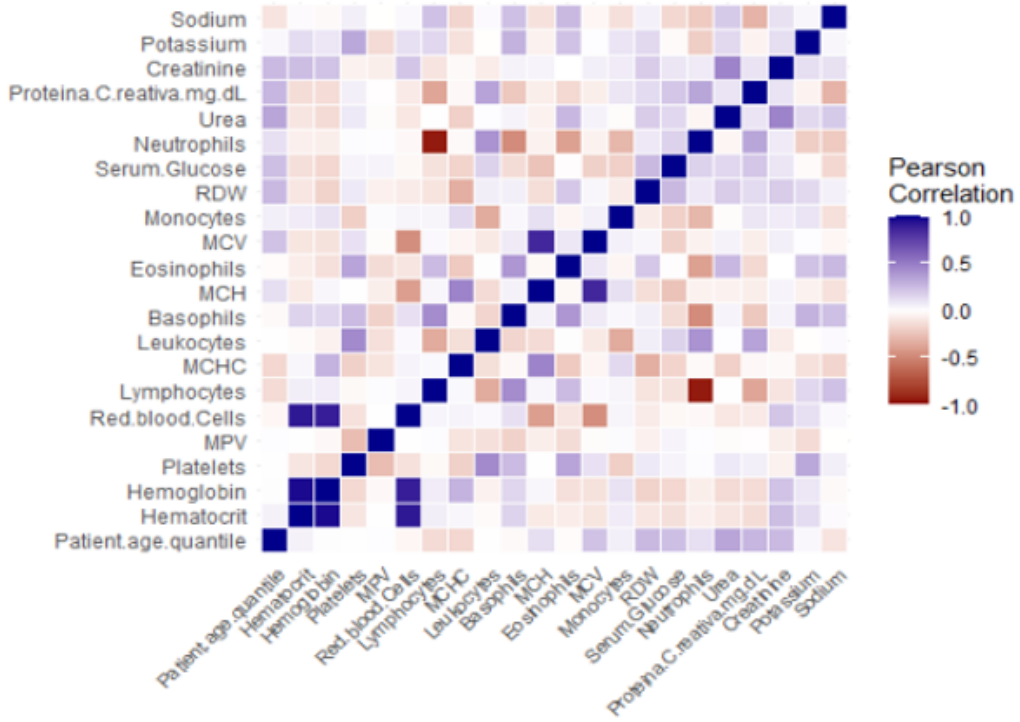

Figura 2.2: O matrice de corelație care exprimă corelațiile dintre o caracteristică și rezultatul testului de COVID-19 pentru eșantionul de date specificat [9].

Următorul articol analizat este *The Outbreak of Coronavirus Disease 2019 (COVID-19) — An emerging global health threat*, cu date inspirate din articolul creat de Kurkina și Koltsova. Pentru toate modelele, AUC a fost semnificativ mai mare de 0.5, ceea ce indică un model bun. Modelul CART a arătat cea mai mare AUC în comparație cu alte modele. Nu a existat o variație apreciabilă între niciunul dintre modelele de predicție folosind criteriile intervalului de încredere de 95%. Modelul CART a avut o putere predictivă semnificativă în comparație cu alte modele, privind metrica AUC. Pentru modelul CART, Modelul de suport vectorial liniar și radial și Rețeaua neuronală densă, aria de sub curba ROC a fost 0.98, 0.86 și, respectiv, 0.8.

În consecință, acest studiu confirmă că modelul CART este un predictor destul de puternic al COVID-19 pe baza caracteristicilor rezultatelor de laborator [18]. Aria de sub curba ROC a fost legată cel mai mult de variabile. Algoritmul CART are cea mai bună valoare deoarece are cea mai mică zonă sub curbă atunci când se formează o linie

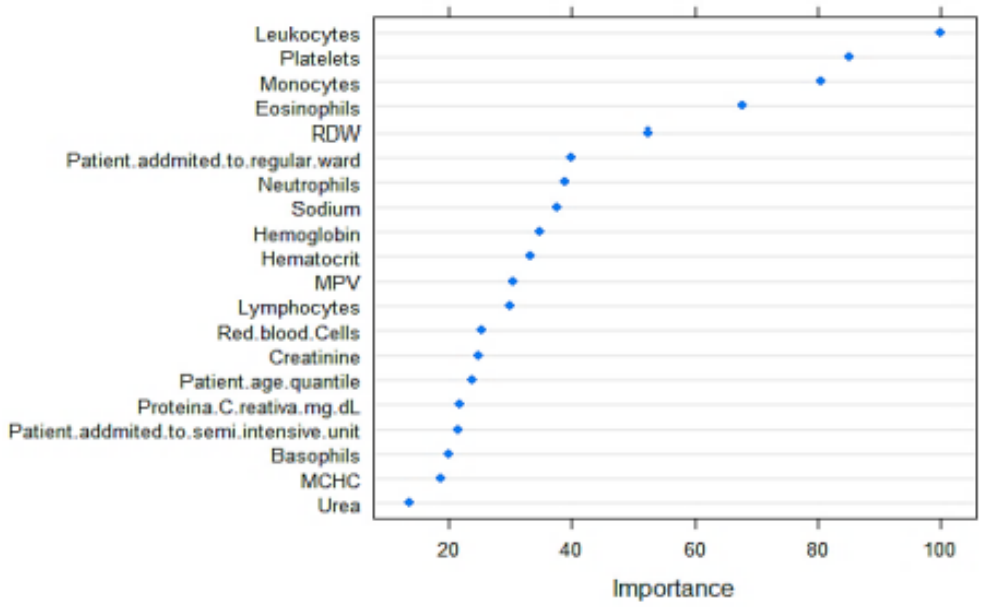

Figura 2.3: Importanța caracteristicilor (variabila de predicție) la *CART* [8].

diagonală, începând de la începutul axei Ox și Oy.

Aceste articole indică faptul că este util să se calculeze metricile de evaluare a performanței modelului, cum ar fi sensibilitatea și specificitatea [6]. Unele articole specifică și că precizia, precum și valoarea predictivă pozitivă și cea negativă sunt alți indicatori utili.

Pe lângă acestea, autorii articolului *COVID-19 Diagnosis Prediction: A Machine Learning Approach* au aplicat cinci algoritmi de învățare automată, implicând rețele neuronale, Pădurea aleatorie, Extreme Gradient Boost (XGBoost): arborii care stimulează un gradient, LR: regresia logistică și SVM: support vector machine, pentru a investiga pericolul de COVID-19 cu diagnostic pozitiv și au luat în considerare cinsprezece caracteristici predictive [7]. Datele au fost colectate de la două sute de pacienți adulți, în Spitalul israelit Albert Einstein din Brazilia [8].

Performanța predictivă a fost măsurată prin calcularea ariei de sub curba ROC (AUC), specificitatea, sensibilitatea, scorul Brier, scorul F1, valoarea predictivă pozitivă și valoarea predictivă negativă. Cea mai bună redare de predicție a fost realizată de algoritmii Modelul cu suport vectorial și Pădurea aleatorie, cu un AUC de 85% pentru setul de testare care a fost de 98%. Modelul XGBoost a fost folosit pentru a prezice rezultatele COVID-19 cu setul pentru teste de aproape 70% și era cel mai bun; modelul Arborelui aleatoriu s-a folosit pentru prezicerea internărilor în spital pentru acei pacienți care sunt pozitivi la virus, unde este inclus ceva mai mult de 90% din date pentru setul de testare. Cea mai benefică metodă este Arborele de clasificare și regresie (CART). În articolul creat de E. S. Kurkina și E. M. Koltsova se observă o acuratețe de 92.5%.

Un alt studiu, *Machine learning applied to diagnosis of human diseases: A systematic review*, a inclus cinci abordări care sunt bine-cunoscute în domeniul de învățare automată, inclusiv arbori de decizie, SVM, rețeaua neuronală artificială (ANN), păduri aleatoare (RF), Arborele de decizie și CART, în timp ce copacii amplificați de gradient au fost considerați la procesul de clasificare cu referire la problematica selectată [9],

pentru care au calculat sensibilitatea și specificitatea pentru a obține un măsurător de performanță.

Într-un alt studiu, *Towards An Artificial Intelligence Framework for Data-Driven Prediction of Coronavirus Clinical Severity*, [10] severitatea clinică a coronavirusului a fost prezisă folosind tehnici de învățare automată. Setul de date a fost obținut de la două spitale din Wenzhou, China. Ei au luat în considerare unsprezece caracteristici predictive și au aplicat clasificatori diferiți, cum ar fi regresia logistică, KNN: K cel mai apropiat vecin, arborele de decizie, pădurile aleatorii și SVM. Clasificatoarele au fost testate luând în considerare doar valorile de precizie. SVM a arătat cea mai bună valoare a preciziei, de 80%.

S-au analizat încă cinci articole conexe cu această lucrare și apropiate cu această tematică și cu virusul respectiv.

Statisticile matricei de corelație pentru fiecare variabilă au fost evaluate utilizând corelații, după caz; un exemplu ar fi coeficientul de corelație *Pearson* [11]. Acest studiu s-a concentrat pe analiza corelației utilizând eticheta clasei. Niște caracteristici au afișat relații puternice între COVID-19, care era eticheta clasei. Se poate observa o matrice de corelație care oferă un diagnostic al conexiunii de perechi dintre fiecare variabilă. Matricea de corelație arată că o rată pozitivă mare care este aproape de 1,0 ce arată o corelație pozitivă puternică, este variabila *hematocrit*. *RDW* și hemoglobina au, de asemenea, o corelație pozitivă puternică. Următorul articol analizat este *The outbreak of Coronavirus Disease 2019 (COVID-19)—An emerging global health threat*, cu date inspirate din articolul creat de Kurkina și Koltsova, *Mathematical modeling and forecasting of the spread of the COVID-19 coronavirus epidemic*.

Pentru toate modelele, aria de sub curbele ROC a fost semnificativ mai mare de 0.5, cum este indicat și în figură. Modelul CART arată cea mai mare zonă de sub curba ROC în comparație cu alte modele de predicție. Acolo nu a existat o variație considerabilă între niciunul dintre modelele de predicție care utilizează criteriile unui interval de încredere de 95%. Chiar și așa, valorile p ale modelului CART și ale tuturor celorlalte modele au fost mai mici decât celelalte comparații. În plus, doar modelul CART a avut o putere de predicție semnificativ mai bună decât altele în funcție de zona de sub curba ROC [8]. Pentru modelul CART, Modelul cu suport vectorial (SVM) liniar și radial, și rețelele neuronale aria sub curba ROC a fost 0.98, 0.86 și 0.8, respectiv.

În consecință, acest studiu confirmă că modelul CART a fost un mare predictor al COVID-19 pe baza caracteristicilor rezultatelor de laborator [12]. Aria de sub curba ROC a fost mai mult legată de variabile.

Ce ar fi de evidențiat la această parte este că se încearcă estimarea importanței variabilei pentru fiecare model utilizat [11]. Cu cât există mai multe variabile importante, cu atât este mai mică aria de sub curba ROC. Totuși, aici se va scădea din 1 valoarea obținută ca și AUC, iar acesta va fi rezultatul final pentru AUC. O concluzie este că algoritmul Arborele de clasificare și regresie are cea mai bună arie deoarece dispune de cea mai mică zonă de sub curbă atunci când se formează o linie diagonală, ce pornește de la începutul axei Ox și Oy.

2.4 Evaluarea modelelor

Evaluările modelelor sunt importante pentru verificarea performanței și eficacității modelelor de învățare automată. Scopul evaluării modelului este de a măsura cât de bine se poate generaliza un model la date noi.

2.4.1 Statistica de descriere

Statistica de descriere, inclusiv mărimi cum ar fi dispersia, deviația standard, media, mediana, cuartilele, diagramele cu casete, histogramele și curbele de densitate, joacă un rol semnificativ în proiectele de învățare automată [30].

Înainte de a începe să se construiască modele de învățare automată, este important să se evalueze calitatea datelor. Statistica de descriere se utilizează pentru a identifica eventualele erori sau anomalii în date, cum ar fi valori lipsă, date extrem de mari sau mici, sau date care diferă semnificativ de restul setului de date. [19] Statistica de descriere poate fi folosită pentru a identifica proprietățile relevante ale setului de date. De exemplu, histograma sau curba de densitate poate arăta dacă distribuția datelor este simetrică sau asimetrică și poate ajuta la determinarea dacă este necesară transformarea datelor [20]. De asemenea, diagramele cu casete pot identifica potențialele outlier-e care pot afecta performanța modelului. Statistica de descriere poate fi folosită pentru a evalua performanța modelelor de învățare automată. De exemplu, deviația standard și media pot fi utilizate pentru a evalua erorile de predicție, iar histograma sau curba de densitate a erorilor poate ajuta la identificarea dacă erorile sunt distribuite normal sau au alte distribuții. Diagramele cu casete și histogramele pot arăta cum modelele clasifică diferite clase în funcție de caracteristicile de intrare. Această statistică este importantă în proiectele de învățare automată deoarece ajută la înțelegerea datelor și la identificarea problemelor care pot afecta performanța modelelor [21]. Folosind această statistică și elementele ei, se poate asigura precizia și robustețea modelelor construite.

2.4.2 Indicatori statistici

Indicatorii statistici sunt măsuri numerice utilizate pentru a evalua relațiile dintre variabile. Indicatorii statistici care exprimă calitatea modelelor, calculați în experiment, sunt eroarea medie pătrată, eroarea medie absolută, coeficientul de determinare și coeficientul de corelație Pearson. Pentru cazurile implicite, particulare, ale setului de date DS2, am calculat aria de sub curba ROC (AUC). Ce reprezintă fiecare indicator?

Eroarea medie pătrată (MSE) este un estimator al mediei pătratelor erorilor. Reprezintă o funcție de risc corespunzătoare valorilor așteptate. Eroarea medie pătrată este întotdeauna una strict pozitivă deoarece dispune de estimatori care nu iau în considerare valorile cele mai acurat estimate [22]. Cum se calculează? *MSE* este suma tuturor diferențelor de pătrat dintr-un set împărțit la numărul de elemente. Diferența se face între elementele observabile, coloanele predictor, și coloana de răspuns [18].

Eroarea medie absolută (MAE) măsoară magnitudinea medie a erorilor dintr-un set de predicție, fără a lua în considerare direcția acestora [23]. Măsoară acuratețea pentru variabile continue. MAE este o măsurătoare de evaluare a modelului utilizată cu modelele de regresie. Eroarea medie absolută a unui model, în raport cu un set de testare, este media valorilor absolute ale erorilor individuale de predicție. Cum

se calculează? MAE semnifică suma diferenței (în valoare absolută) dintre rezultatul actual y și cel obținut după predicție, de pe poziția i, împărțit la numărul de elemente din set [18].

Coeficientul de determinare denotat cu r^2 este proporția variațiilor dintre variabila predictor și cea de răspuns. Aceasta se folosește în cadrul unui model statistic al cărui scop este predicția viitoarelor rezultate conform ipotezelor bazate pe informații similare. Acest coeficient este mai informativ decât eroarea medie pătratică și eroarea medie absolută [8].

Coeficientul de corelație măsoară rezistența și direcția relației liniare dintre două variabile. Acesta variază de la -1 la 1, unde -1 indică o corelație negativă perfectă, 0 nu indică nicio corelație și 1 indică o corelație pozitivă perfectă [24]. Coeficientul de corelație este folosit în mod obișnuit pentru a evalua puterea relației dintre două variabile și poate ajuta la identificarea modelelor în date.

Având în vedere analiza setului de date COVID-19, performanța modelelor predictive, care identifică zonele cu riscuri de COVID-19 pe baza mai multor variabile, poate fi evaluată folosind AUC [19]. Aceasta poate fi folosită pentru a compara performanța diferitelor modele și poate oferi informații despre cât de bine modelul poate discrimina țările cu risc de COVID-19 și cele fără risc.

2.4.3 Metrici de evaluare a performanței

Metricile de evaluare a performanței modelelor, folosite în acest studiu, sunt acuratețea, sensibilitatea, specificitatea, precizia și scorul F1 [14].

Clasificarea realizată este una binară, iar limita raportului care clasifică țările în 1) grupul de țări cu risc și 2) țări fără risc, este 0.1 (10% din populație) pentru setul de date despre Oceania, și 0.003 (adică 0.3% din populație) pentru ultimele 100 de țări de pe *Worldometer Online*, până în septembrie 2020. A doua clasificare binară este realizată folosind numai cazurile totale (zonele cu risc: $\geq$ 40000 cazuri totale pentru setul despre Oceania; zonele cu risc: $\geq$ 4000 cazuri totale pentru setul cu ultimele 100 observații de pe Worldometer).

Cum se definește fiecare metrică individual? Acuratețea este gradul de apropiere dintre măsurătoare și valoarea sa adevărată. În proiectul de față, se calculează exactitatea unui model cu privire la numărul de infectați față de populația țării indicați de model și numărul adevărat de infectați față de populația țării alese: valorile de pe *Worldometer*. Pe de altă parte, sensibilitatea este procentul de *True positives* din toate verificările pozitive. Pe exemplul acestui proiect, sensibilitatea este probabilitatea ca țara care într-adevăr este una cu risc de rată mare de infecție cu COVID-19 să fie pe lista cu toate țările cu risc. Mai precis, se calculează proporția dintre numărul total de cazuri, considerând luna septembrie din 2020, respectiv luna martie din 2022, și numărul populației pentru țara selectată. Specificitatea reprezintă probabilitatea țărilor care se află în grupul fără risc de infecție mare, conform modelului, să fie într-adevăr țări cu risc scăzut sau la limită. Precizia este un măsurător care indică cât de apropiate sunt două măsurătoare una de cealaltă. Scorul F1 reprezintă media armonică dintre sensibilitate și precizie.

Deoarece în proiect s-a construit o matrice de confuzie, apelând funcția *confusion_matrix(y_test, y_pred)* unde lista de testare și cea de predicție pentru variabila dependentă sunt introduse ca si argumente. Pentru a obține lista cu valorile predictive, a trebuit să se obțină proporția dintre coloana *Total Cases* și *Population*. S-au

construit valori de 0 și 1 pentru valorile care nu depășesc, respectiv care depășesc limita (apropiată de medie) indicată: 0.003 pentru primul set de date setul de date cu ultimele 100 țări de pe Worldometer și 0.1 pentru setul de date despre țările din Oceania.

Matricea de confuzie este construită pentru fiecare model în parte. Pentru fiecare model se va obține o listă diferită de valorile predictive. Din această matrice se extrag valorile pentru True Positive, True Negative, False Positive, False Negative [25]. Conform acestora, s-au calculat metricile de evaluare a performanței. Cu acest rost, formulele pentru fiecare metrică vor fi oferite în continuare.

Sensibilitatea arată proporția de rezultate pozitive reale care sunt identificate corect de un model de clasificare [26]. Formula este următoarea:

$$sensibilitate = \frac{TP}{TP + FN}. \tag{2.1}$$

Specificitatea arată proporția de rezultate negative reale care sunt identificate corect de un model de clasificare ales. Formula este:

$$specificitatea = \frac{TN}{TN + FP}. \tag{2.2}$$

Acuratețea arată proporția de rezultate corect identificate de un model de clasificare, indiferent de valoarea pozitivă sau negativă a lor [27]. Formula pentru acuratețe este următoarea:

$$acuratețe = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.3}$$

Precizia este o măsură statistică care arată proporția de rezultate pozitive adevărate în raport cu totalul de cazuri clasificate ca pozitive. Formula este următoarea:

$$precizie = \frac{TP}{TP + FP}. \tag{2.4}$$

Scorul F1 este o măsură statistică care combină precizia și sensibilitatea unui model de clasificare într-un singur scor. Este folosit pentru a evalua performanța unui model de clasificare în cazul în care nu există un echilibru între cazurile pozitive și cazurile negative [28]. Formula este:

$$scor\ F1 = \frac{TP}{TP + \frac{1}{2}(FN + FP)}. \tag{2.5}$$

S-a obținut câte o diagramă pentru fiecare metrică de evaluare a performanței, pentru fiecare model. Rezultatele sunt reprezentate într-o singură figură.

Capitolul 3

Materiale și metode

În acest capitol vom discuta despre seturile de date selectate și despre clasele și funcțiile utilizate în acest proiect. Acei termeni și concepte de demers științific, stabilite, notate și definite anterior, în prezentul capitol sunt concretizate prin termeni de programare, cod și exemple, prin structura secvențială a celor zămislite și evidențiate teoretic.

3.1 Selecția seturilor de date

Atunci când dezvoltăm un proiect de învățare automată folosind date COVID-19, alegerea setului de date adecvat este un fapt vital. Selecția poate influența acuratețea modelelor, relevanța datelor prezise și diversitatea sau uniformitatea performanței modelului [29]. Am creat două seturi de date pe baza unor seturi de date găsite pe platforma Kaggle. Datele despre țările din Oceania sunt colectate într-un set de date până în martie 2022, când am descărcat întregul conținut de pe Kaggle. Al doilea set de date este despre ultimele 100 de țări de pe Worldometer (ordonate după numărul total de cazuri de infectați cu SARS-CoV-2), până în septembrie 2020. Setul de date pentru țările din Oceania (DS1) a avut 10 observații, însă a fost mărit la 110 observații, pentru a avea suficiente date de antrenare. La fel s-a procedat și cu ultimele 100 de țări de pe Worldometer (cazuri totale până în septembrie 2020) (DS2), care a ajuns la 1100 de observații. Am realizat augmentarea setului utilizând o funcție de sumă cumulativă pentru coloanele Total Cases și Population. Am executat aceeași procedură de unsprezece ori, la ambele seturi de date. Unele observații din cadrul de date Pandas sunt similare, însă nu și la fel. Am și incrementat valorile acestor două coloane. Setul de date (cu actualizări zilnice) pentru construirea DS2 poate fi găsit aici [1], în timp ce întregul conținut pentru DS1 (cu actualizări zilnice) aici [2]. Din setul de date asociat cu setul de date DS2, am selectat doar ultimele 100 de țări din toate țările (ordonate după numărul total de infectați); din setul de date asociat cu DS1, am tratat doar 10 țări din 17. Ideea principală este să vedem dacă ultimele 100 observații pot fi date bune de antrenare, astfel încât modelele noastre să prezică bine pentru primele 100 de țări din listă. Motivul selecției setului de date DS1 este acela de a afla dacă modelele noastre pot fi predictori buni pentru țările insulare și peninsule.

[1] https://www.kaggle.com/datasets/selfishgene/covid19-worldometer-snapshots-since-april-18?resource=download

[2] https://www.kaggle.com/datasets/anandhuh/covid-in-oceania-latest-data

3.2 Prelucrarea datelor

Alți oameni implicați în cercetare pot replica rezultatele și pot confirma exactitatea constatărilor de față, oferind o explicație a metodelor de prelucrare a datelor. În cazul seturilor de date COVID-19, în care calitatea și fiabilitatea datelor sunt esențiale, acest lucru este deosebit de semnificativ. Descrierea explicită a etapelor de prelucrare a datelor face ca proiectul să fie transparent, permițând altora să înțeleagă tehnica și să judece calitatea datelor [30].

Bibliotecă numită Pandas este folosită pentru transpunerea fișierelor CSV în cadre de date (dataframes), iar pentru procesarea ulterioară a datelor, folosim funcția *read_csv()*. Înregistrările sunt afișate cu funcția *head()*, afișând numele coloanei și datele acesteia. Valorile lipsă și aberante sunt completate cu media valorilor din acea coloană [de exemplu, $df = df.fillna(df.mean())$]. Îndepărtăm coloana *Total Cases* și alte coloane necorelaționale din cadrul de date inițial, pentru a obține un cadru de date în care se află potențialele variabile de predicție. După aceea, facem niște procesări de bază, cum ar fi valori minime, maxime, medii (mean()), sau mediana, dispersia, abaterea standard, cuantile și cuartile ale coloanelor *Total Recovered*, *Active Cases*, *Population*, *Total Cases*, etc.

	Total Tests
0.25	11,071.5
0.5	47,732
0.75	156,697.1765

Figura 3.1: Cuartilele pentru coloana *Total Tests* a setului de date DS2.

Trecem la etapa de modelare a datelor. Îndepărtăm datele care nu sunt necesare atunci când modelăm variabila predictor. Reprezentăm rezultatele grafic; Afișăm histogramele, diagramele cu casete și curbele de densitate pentru anumite coloane.

Se calculează coeficientul de corelație Pearson între variabila predictor selectată și variabila răspuns, *Total Cases*. În total, există cinci coloane care sunt corelate pozitiv cu variabila răspuns. Coeficienții de corelație sunt afișați prin intermediul unei hărți termice (`heatmap()`) din biblioteca Seaborn. Prin selectarea coloanelor necesare pentru variabila independentă, se poate forma un model liniar. Un model liniar este creat cu clasa `LinearRegression` din modulul *linear_model* din librăria Scikit-learn.

3.3 Construirea indicatorilor statistici

Clasele utilizate sunt `KNeighborsRegressor` și `KNeighborsClassifier` din modulul *neighbors* al Scikit-learn, `DecisionTreeRegressor` și `DecisionTreeClassifier` din modulul *tree*, SVR și SVC din modulul *svm*, și `XGBRegressor` și `XBClassifier` dintr-o bibliotecă numită *xgboost*. Am folosit regresori la eroarea medie pătratică (MSE) și eroarea medie absolută (MAE). Fiecare model are minim un parametru care variază, cu scopul de a obține MSE și MAE pentru 40 de valori diferite. Pentru modelul K-Nearest Neighbors, parametrul *k_neighbors* variază. Pentru modelul Decision Tree

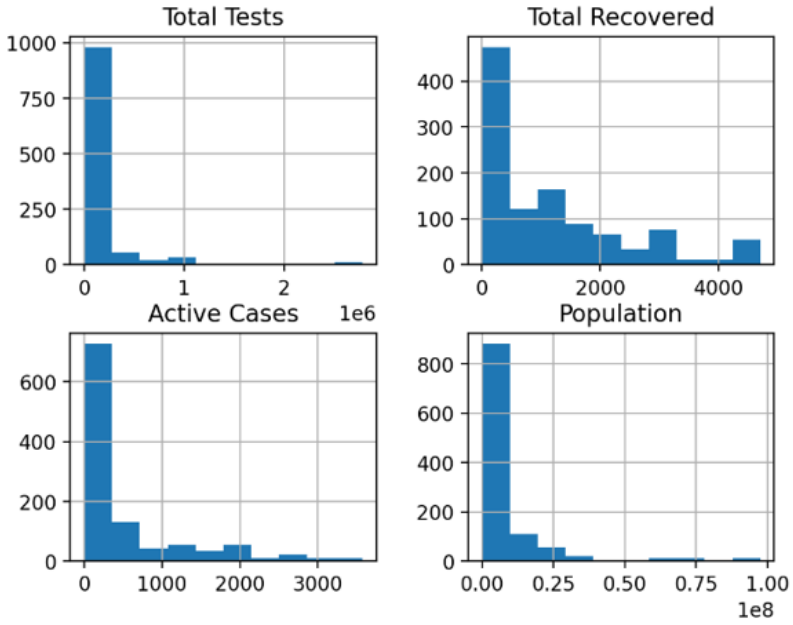

Figura 3.2: Patru histograme pentru setul de date DS2.
Total Tests: Histograma pentru coloana *Total Tests*.
Total Recovered: Histograma pentru coloana *Total Recovered*.
Active Cases: Histograma pentru coloana *Active Cases*.
Population: Histograma pentru coloana *Population*.

(DT) (*criterion='entropy'*), parametrul *max_depth* variază, în timp ce pentru Arborele de clasificare și regresie (*criterion='gini'*), care are aceeași clasă ca DT, pe lângă *max_depth* variază și min_leaf. Pentru Modelul cu suport vectorial, parametrul de regularizare C este variabil. În cele din urmă, pentru modelul Extreme Gradient Boost, subeșantionul ($subsample = [0, 1]$) este parametrul variabil. Pentru fiecare parametru variabil, există un interval de $[1, 40]$. Pentru a obține MAE și MSE, folosim funcțiile `mean_squared_error()` și `mean_absolute_error()` din modulul de metrici al Scikit-learn. În aceste funcții, specificăm variabila dependentă a testării și valoarea prezisă a acesteia, la fiecare pas. Clasificatorii sunt utilizați pentru coeficienți și pentru modelele construite pe setul de antrenare și testare (funcția `train_test_split()`). Folosim funcțiile `score()` și `pearsonr()` pentru acești indicatori statistici. Pentru indicatorul AUC, folosim funcția `roc_auc_score()`. Valorile MSE și MAE, împreună cu coeficienții de determinare și corelare, sunt afișate (`plot()`) și salvate (`savefig()`) într-un format PDF prin modulul din Matplotlib numit *pyplot*.

3.4 Construirea metricilor de evaluare

Propunem două criterii de clasificare a modelelor: 1) raportul dintre totalul de cazuri și populație, și 2) valoarea pentru Total Cases.

Pentru primul criteriu, creăm o nouă coloană (numită $Target$), care reprezintă raportul dintre valorile coloanelor Total Cases și Population. Apoi, calculăm o valoare medie pentru noua coloană, pentru ambele seturi de date. Obținem 0.003 pentru DS2 și 0.1 pentru setul de date DS1. Astfel, creăm clase pentru clasificarea binară

pe baza acestor limite. Pentru fiecare metodă, ar trebui să transformăm datele coloanelor astfel încât acestea să fie compatibile. Pentru această intenție, folosim metoda $fit\_transform(y)$ a clasei LabelEncoder, unde y este variabila răspuns. Pentru a calcula valorile de evaluare (sensibilitate, specificitate, acuratețe, precizie și scorul F1), inițializam cinci liste cu toate zerourile, unde fiecare listă este asociată cu o măsurătoare. Folosim clasa StandardScaler pentru a transforma setul de antrenare și test al variabilei independente. Următorul pas este ajustarea datelor și estimarea variabilei dependente. După aceea, creăm o matrice de confuzie printr-o funcție, `confusion_matrix()`. Această matrice de confuzie ne permite să extragem valorile predictive pozitive și negative: Adevărat Pozitiv, Fals Pozitiv, Adevărat Negativ și Fals Negativ. Folosim aceste valori cu scopul de a construi valorile noastre de evaluare a performanței modelelor, folosind formule specifice. În cele din urmă, punem valoarea obținută pentru fiecare metrică în lista sa. O diagramă cu bare este creată prin funcția `bar()` din modulul *pyplot*. Creăm o legendă cu modele, le salvăm într-un fișier PDF și le afișăm [31].

Al doilea criteriu este delimitarea zonelor de risc de zonele sigure printr-o valoare *Total Cases*. Calculăm media acestei coloane și o punem ca limită. Pentru setul de date DS2, această limită este 4000, în timp ce pentru setul de date DS1 este 40000. Aceleași proceduri și metode (ca la prima clasificare pe criterii) sunt aplicate pentru această clasificare binară. Pentru acuratețe, indicăm cinci porțiuni de date, folosind metoda de validare încrucișată, pentru a afla care modele de învățare automată sunt cele mai bune pentru unele părți individuale ale setului de date (DS1 și DS2). Rezultatele diferă de la un criteriu la altul.

3.5 Diagrama de sistem pentru construirea modelelor de învățare automată

Această secțiune o să ofere o descriere detaliată a abordării tehnice utilizate pentru dezvoltarea și evaluarea modelelor.

Această secțiune ar trebui să acopere următoarele aspecte:

- Colectarea și pregătirea datelor: Descrierea seturilor de date COVID-19 utilizate în studiu, cum au fost colectate și cum au fost pregătite pentru utilizare în modelele de învățare automată.
- Selectarea caracteristicilor: Se menționează și analizează caracteristicile care au fost selectate pentru a fi utilizate în modelele predictive, inclusiv modul în care au fost alese și orice pași de preprocesare care au fost efectuați.
- Algoritmi de învățare automată: Se discută algoritmii de învățare automată utilizați pentru a dezvolta modelele predictive. Se explică de ce au fost aleși acești algoritmi și se descrie orice modificare sau reglare a parametrilor.
- Evaluarea modelelor: Se detaliază procesul utilizat pentru a evalua performanța modelelor predictive. Se discută valorile utilizate pentru evaluarea modelelor și explică cum au fost selectate acestea.

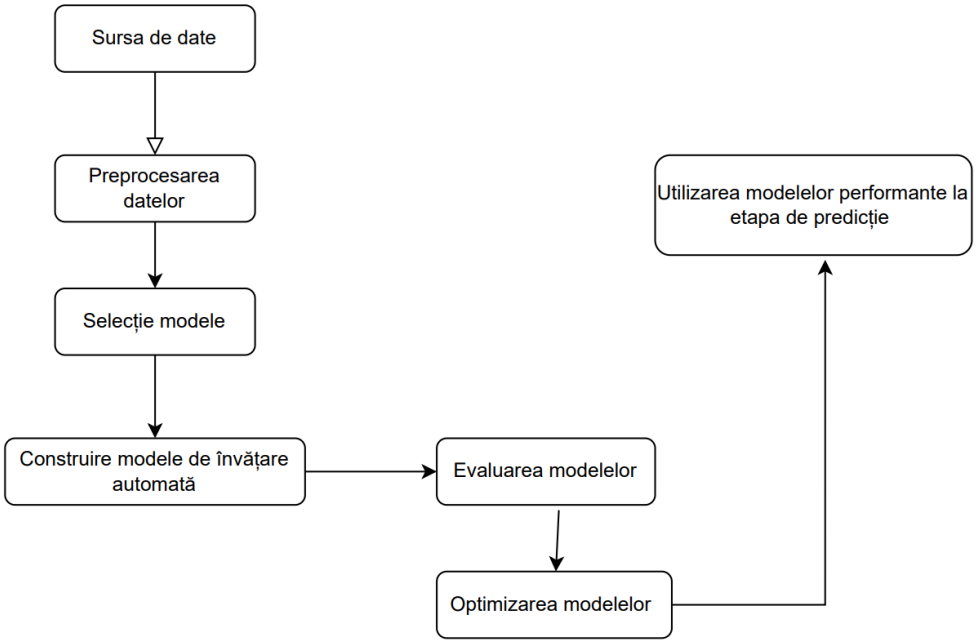

Figura 3.3: Diagrama de context a sistemului

- Rezultate și discuții: Se prezintă rezultatele studiului, inclusiv performanța modelelor predictive și orice informații obținute în urma analizei. Se discută limitele studiului și se sugerează domenii pentru cercetările viitoare.

Se vor combina întrebările/ipotezele de cercetare abordate în capitolele anterioare. Acest capitol ar trebui să ofere o imagine de ansamblu asupra abordării utilizate pentru dezvoltarea și evaluarea a câteva modele predictive de bază, evidențiind principalele caracteristici și algoritmi utilizați. În cele din urmă, ar trebui să previzualizeze pe scurt rezultatele și concluziile/părerile critice.

3.6 Aplicație *Streamlit*

La etapa de urcare a aplicației pe Streamlit Cloud este nevoie de un fișier denumit requirements.txt unde să se indice librăriile care să fie automat instalate odată cu rularea aplicației. Mai este nevoie și de un fișier config.toml unde să se indice portul pe care se rulează aplicația. Apoi, este nevoie și de un fișier setup.sh unde să se indice proprietățile din fișierul cu extensia .toml. Ultimul fișier necesar este Procfile, unde se specifică tipul aplicației: web, fișierul sh și aplicația fișierul Streamlit care o să pornească aplicația. Totuși, nu toate paginile web funcționează în timp util. Sunt multe prelucrări de date la partea de comparație a modelelor de învățare automată și la partea de calcul a timpilor de execuție în ceea ce privește implementarea modelelor de predicție pentru setul de date ales, folosind modelele alese.

S-au introdus toate fișierele și directoarele necesare pe repozitoriul de pe GitHub, iar acestea s-au accesat din Streamlit Cloud. Mai întâi s-a creat un cont pe acea platformă, după care aplicația nou creată s-a legat cu codul de pe GitHub. Trebuia introdus numele repozitoriului, selectat branch-ul (main) și aleasă pagina principală, cea de start, *The_application.py*. Aplicația este disponibilă la următorul link https://fortunab-ml-modelsanalysis-application-zo1ng4.streamlitapp.com. La început,

când aplicația s-a urcat pe Streamlit Cloud, a trebuit foarte mult până să se încarce paginile, în special Regression models și ML models comparison. Problema era că se folosea o librărie *streamlit option menu* pentru stilizarea sidebar-ului. Aceasta s-a eliminat. Apoi, la calculul timpilor de execuție ai algoritmilor, s-a apelat de mai multe ori aceeași funcție, fără a mai fi neapărat nevoie de acea apelare. Acestea s-au eliminat din cod. La fel s-a procedat și la calculul sensibilității, al specificității, al preciziei și al scorului F1. Fiecare metodă, aferentă calculului anume, a fost folosită și la partea de asignare a valorii unei variabile. După ce metodele s-au apelat, o singură dată fiecare, s-a salvat rezultatul în formatul .png. Deci, s-a obținut câte o imagine pentru fiecare metrică de evaluare a performanței modelului.

Structurarea de pagini. Designul de pagini

La prima pagină, Application, s-a construit un footer care precizează cine este autorul proiectului sau al aplicației. Fiecărei pagini s-a adăugat câte o iconiță și o denumire care să fie intuitivă utilizatorilor. Când se dă clic pe pagină, se observă modificarea rutei și a iconiței. Rutele și iconițele paginii s-au creat în așa fel încât s-au specificat denumirile paginilor în noul director, cel cu numele Pages. S-a folosit funcția din Streamlit: set_page_config() pentru a introduce titlul paginii, proprietatea de extindere a barei verticale.

Aplicație disponibilă pentru toate dispozitivele cu sistem *Android*

Folosind tehnologia WebView, și proprietățile acesteia, s-a introdus link-ul aplicației urcate pe Streamlit Cloud și s-a construit o aplicație responsive, disponibilă prin intermediul unui Wi-Fi pe oricare dintre dispozitive, inclusiv telefon mobil și tabletă. Totuși, acestea trebuie să dispună de sistemul de operare Android. Aplicația a fost testată și pe telefonul mobil și pe tabletă și au fost efectuate câteva modificări care implică partea de responsive a aplicației. Acum funcționează bine aplicația și pe telefon. Fișierul *.apk* a fost descărcat, iar aplicația instalată pe telefon.

3.7 Fluxul aplicației dezvoltate

S-a pornit aplicația folosind cadrul de lucru *Streamlit* care nu are nevoie de fiecare dată de tehnologiile web, fiindcă este o tehnologie care dispune de o reprezentare web. Pentru construirea footer-ului a fost nevoie de o secvență de cod HTML și CSS. S-a integrat codul din *PySpark* în Streamlit. Local s-a obținut o interfață web interactivă, unde utilizatorul selectează modelul, după care se returnează datele dintr-o secțiune, metricile de evaluare a performanței și indicatorii statistici pentru o metodă. De asemenea, utilizatorul poate introduce manual în TextFields datele pentru o țară sau un teritoriu autonom cu scopul de a se returna o valoare care indică Total Cases pentru acestea.

În această secțiune va fi descris fluxul aplicației web unde se află rezultatele aplicației.

Au fost efectuate modificări ale paginii cu scopul de a obține o aplicație mai structurată decât ce a fost anterior, din punctul de vedere al esteticii și al accesului. Nu se mai află butoane de tip radio proiectate cu CSS. După ce s-a dat clic pe aceste butoane, într-adevăr s-a vizualizat o altă funcționalitate a aplicației, însă ruta paginii era

aceeași. Cu acest rost, s-au introdus unele metode și prelucrări de date în paginile cum sunt: Dataset processing, Basic graphs, Regression models, ML models comparison.

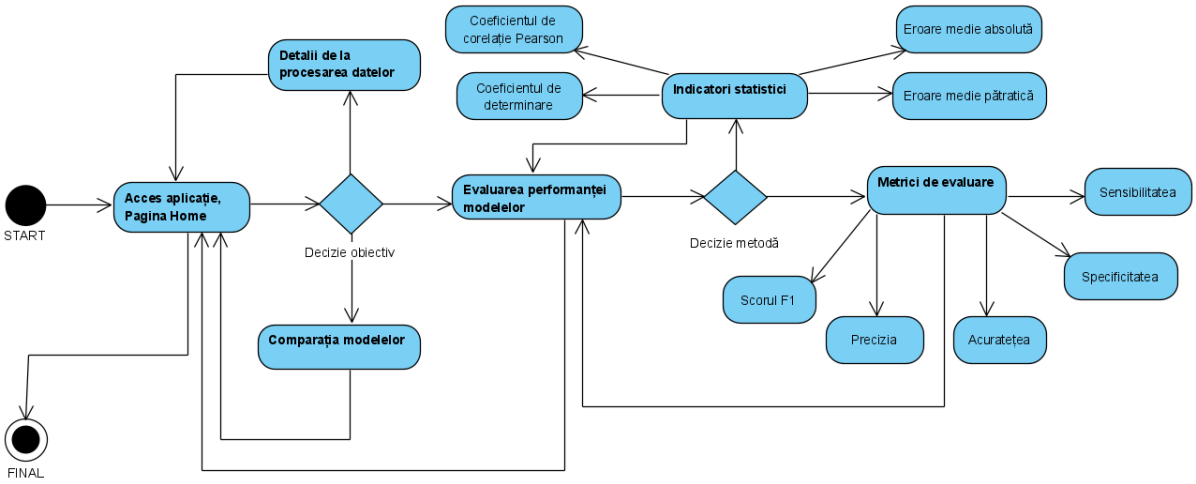

Figura 3.4: Diagrama de acțiune pentru fluxul studiului și a aplicației aferente studiului

O prelucrare a datelor este reprezentarea grafică a acurateței, exactității, modelelor de învățare automată pentru cinci porțiuni de date obținute folosind metoda de validare încrucișată. Această reprezentare grafică se modifică după ce se modifică și setul de date ales pentru prelucrare. Există două seturi de date: setul cu ultimele 100 de țări de pe Worldometer culese până în septembrie 2020 și setul de date despre țările din Oceania până în martie anul acesta. O altă schimbare ar fi folosirea tabelului unde se introduc manual datele despre o țară sau un teritoriu autonom (Denumirea, populația, numărul total de teste, numărul total de recuperați, numărul de critici și cazurile active). Încă o modificare care se evidențiază este utilizarea diferitelor rezultate în cadrul paginii Dataset processing. Aici este afișat numărul de observații pentru observațiile care au valoarea pentru Total Cases mai mică, respectiv mai mare, decât patru mii. Este afișată și media tuturor testelor ale tuturor țărilor. Se observă și un tabel cu cazurile totale aferente unei țări sau unui teritoriu autonom extras pe platforma Worldometer online.

Este reprezentată acuratețea pentru țările care nu reprezintă risc pentru a depăși rata de infecție indicată ca și periculoasă, pentru Oceania. Aici în loc de 0.1% se indică o proporție dintre *Total Cases* și *Population* de 0.05%. Există 5 porțiuni de date, obținute prin metoda de validare încrucișată. Cel mai bun rezultat îl au modelele KNN și CART, și acestea merită a fi considerate relevante pentru cazul în care se tratează țările fără risc la o rată de infecție mare. Pentru a cincea porțiune de date, cel mai bun model este XGBoost, cu o acuratețe de peste 80%.

3.8 Implementare

Următoarea secțiune a acestui capitol cuprinde implementarea aplicației și a elementelor științifice. În cadrul acestei secțiuni vor fi transferate și indicate liniile de cod fundamentale procesului de implementare și dezvoltare a proiectului. După ce sunt introduse în lucrare, acestea vor fi documentate. Partea pricipală a documentației realizate pe bază de cod, va fi accentuată doar datorită funcționalităților esențiale oferite.

Secțiunea de cod ar trebui să ofere cititorului toate detaliile necesare pentru a reproduce rezultatele. Aceasta înseamnă includerea codului de programare folosit pentru a antrena modelele, pentru a efectua preprocesarea datelor și pentru a efectua orice analiză sau vizualizare. Pentru aceasta, trebuie să se ofere o imagine de ansamblu clară a codului. Se începe prin a introduce secțiunea de cod și printr-o prezentare generală la nivel înalt a codului folosit, cum ar fi limbajele de programare și orice biblioteci sau pachete relevante. Se vor includeți fragmente de cod pentru toate părțile cheie ale analizei, cum ar fi preprocesarea datelor, instruirea modelului și evaluarea acestuia, pentru fiecare model individual. Dacă procedura este aceeași, se va menționa că partea respectivă de cod apare de mai multe ori în aplicație, cu mici alterații. Codul este și comentat, iar numele variabilelor sugerează tendința programatorului. Se indică instrucțiuni clare pentru rularea codului; ce pachete și biblioteci trebuie instalate, ce linii de cod se introduc în linia de comandă, care sunt pașii de rulare și cel de pornire a aplicației. Sunt incluse și detalii despre fișierele de intrare necesare. Trebuie specificate și exemple de rezultate generate, cum ar fi tabele, figuri și grafice. Utilizatorul trebuie să se asigure că utilizează versiunea corectă. De asemenea, trebuie să existe și un control al versiunii, cum ar fi *Git*, pentru a urmări modificările aduse codului. Cu acest rost, se va atașa un link către depozitul în care este găzduit codul și unul către aplicație. Scopul secțiunii de cod este de a face analiza transparentă și reproductibilă. Codul ar trebui să fie bine documentat, ușor de înțeles și să poată fi rulat de oricine dorește să reproducă rezultatele.

Prima secvență de cod care ar trebui să se evidențieze importă mai multe biblioteci de învățare automată în Python, și anume:

```
from sklearn.neighbors import KNeighborsClassifier,
    KNeighborsRegressor
from sklearn import tree
from sklearn.svm import SVC, SVR
from xgboost import XGBClassifier, XGBRegressor
```

- biblioteca *sklearn.neighbors* care include clasele *KNeighborsClassifier* și *KNeighborsRegressor*. Aceste clase sunt folosite pentru a crea modele care fac o clasificare sau predicție cu variabile țintă continue pe baza celor mai apropiați vecini din spațiul de caracteristici.

- biblioteca *sklearn.tree*, de unde se importă modulul *tree*. Acest modul este folosit pentru a crea arbori de decizie pentru probleme de clasificare sau regresie. De asemenea, acest algoritm se poate optimiza pentru a ajunge la un model și mai performant: regresie și clasificare folosind modelul CART.

- biblioteca *sklearn.svm* care conține clasele *SVC* și *SVR*. Aceste clase sunt folosite pentru a crea modele de SVM pentru sarcini de clasificare sau regresie.

- biblioteca *xgboost* care dispune de clasele XGBClassifier și XGBRegressor. Aceste clase sunt folosite pentru a crea modele de creștere a gradientului pentru probleme de clasificare sau regresie.

Pentru a utiliza aceste biblioteci, de obicei se creează o instanță a clasei dorite, se setează parametrii necesari, se face o potrivire (fitting) a modelului la datele selectate, adunate, folosind un set de antrenament și apoi folosind modelul pentru a face predicții asupra datelor noi.

Apoi, urmează o secvență de cod care îndeplinește mai multe sarcini legate de analiza datelor și învățarea automată.

```
# Citire si afisare date
df = pd.read_csv('sample_data/septembrie2020_augmentat.csv')
print(df.head(10).to_string(index=False))
df.head()
df.info()

# Tratare coloane nil
df.isnull().sum()
df = df.fillna(df.mean())
df.isnull().sum()
print(df.head())

def corelatie():
    # print("\nVerif corelatia Pearson dintre variabile si Total Cases")
    print("\nCorelatia intre variabilele independente si cea dependenta ")
    for i in X.columns:
        corelatie, _ = pearsonr(X[i], y)
        print(i + ': %.2f' % corelatie)
corelatie()

def medie_dispersie_devstd_Population():
    # medie coloana Population
    pavg = df["Population"].mean()
    # dispersie coloana Population
    pv = df["Population"].var()
    # deviatie standard coloana Population
    psd = df["Population"].std()
    # mediana coloana Population
    pmed = df["Population"].median()
    # cuartila coloana Population
    pq = df["Population"].quantile([0.25, 0.5, 0.75])
    return pavg, pv, psd, pmed, pq
medie_dispersie_devstd_Population()
```

Citire și afișare date: Această secțiune a codului citește dintr-un fișier CSV numit „septembrie2020_augmentat.csv” folosind biblioteca Pandas și stochează datele într-un Pandas DataFrame numit *df*. Apoi tipărește primele 10 rânduri ale DataFrame-ului folosind metoda *head()* și afișează informații suplimentare despre DataFrame folosind metoda *info()*.

Tratare coloane nil: Această secțiune a codului verifică valorile nule în DataFrame folosind metoda *isnull()* și apoi înlocuiește orice valoare nulă cu valoarea medie a coloanei folosind metoda *fillna()*. Apoi tipărește din nou primele câteva rânduri ale DataFrame-ului pentru a confirma că nu mai există valori nule.

Vectorul de coloane pentru variabila independentă și specificație variabilă dependentă: Această secțiune a codului creează două variabile noi, X și y, pentru a reprezenta variabila independentă și, respectiv, dependentă. Variabilele independente sunt create prin eliminarea coloanelor „Country” și „Total Cases” din DataFrame folosind metoda *drop()*. Variabila dependentă este creată prin selectarea numai a coloanei „Total Cases” din DataFrame.

Metoda pentru coeficienții de corelație la nivel general: Această secțiune definește o

nouă funcție numită *corelatie()* care calculează coeficientul de corelație Pearson între fiecare coloană din variabilă independentă din X și variabila dependentă y. Acest lucru se face folosind metoda *pearsonr()* din biblioteca *SciPy*. Coeficienții de corelație rezultați sunt afișați în consolă.

O altă funcție, `medie_dispersie_devstd_Population()`, este definită și calculează media, varianța, abaterea standard, mediana și cuartilele pentru coloana „Populație" din DataFrame. O altă funcție, `sumar_toate()`, calculează aceleași statistici pentru orice coloană dată. Următoarele două funcții creează matrici de hărți termice, heatmaps, pentru a vizualiza corelația dintre toate coloanele din DataFrame, respectiv doar variabila independentă (X).

În cele din urmă, codul selectează un subset de variabile independente (Total Tests, Total Recovered, Serious or Critical, Active Cases) și le atribuie la X (`X = df[['Total Tests', 'Total Recovered', 'Serious or critical', 'Active Cases']]`). Alte două librării sunt folosite în proiect, pentru vizualizare: *matplotlib* cu modulul *pyplot* și librăria *seaborn*.

Mai departe, codul definește mai multe modele de învățare automată pentru sarcini de predicție, fiecare cu parametri și valori de evaluare diferiți. Se prezintă un scurt rezumat al fiecărei funcții:

- model_DT_coeffs: Clasificator de arbore de decizie cu criteriu de entropie și adâncime maximă de 2. Returnează metrici de evaluare R-pătrat și RMSE.
- model_DT_msq_mabs_e: Clasificator de arbore de decizie cu criteriu de entropie și adâncime maximă de 5. Returnează etichetele estimate, eroarea pătrată medie și valorile de evaluare a erorii medii absolute.
- model_KNN_coeffs: Clasificator K-Nearest Neighbours cu 10 vecini. Returnează valorile de evaluare R-pătrat și RMSE.
- model_KNN_msq_mabs_e: Clasificator K-Nearest Neighbours cu 10 vecini. Returnează valorile de evaluare a erorii medii pătrate și a erorilor medii absolute.
- model_SVM_coeffs: Suport Vector Machine Classifier cu nucleu polinomial de grad 3 și formă de funcție de decizie unu-vs-unu. Returnează valorile de evaluare R-pătrat și RMSE.
- model_SVM_msq_mabs_e: Suport Vector Machine Classifier cu nucleu polinomial de gradul 5 și formă de funcție de decizie unu-vs-unu. Returnează valorile de evaluare a erorii medii pătrate și a erorilor medii absolute.
- model_CART_coeffs: Clasificator al arborelui de clasificare și regresie (CART) cu criteriul gini, adâncimea maximă de 3 și mostre minime de frunze de 1000. Returnează valorile de evaluare R-pătrat și RMSE.
- model_CART_msq_mabs_e: Clasificator CART cu criteriul gini, adâncimea maximă de 3 și mostre minime de frunze de 1000. Returnează valorile de evaluare a erorii medii pătrate și a erorilor medii absolute.
- model_XGBoost_coeffs: Clasificator de creștere a gradului extrem cu sub-eșantion de 0,15 și adâncime maximă de 2. Returnează etichetele estimate, scorul de precizie, valorile de evaluare și o analiză de clasificare.

- model_XGBoost_msq_mabs_e: Clasificatorul pentru XGBoost permite calculul erorii medie absolute, respectiv pătratice, folosind caracteristici similare cu cele menționate pentru metoda trecută.

Urmează specificarea erorilor medii pătratice, respectiv absolute, precum și calculul valorilor R-squared și a coeficientului de corelație pentru fiecare model. S-au folosit prelucrări de date pentru 40 de versiuni diferite, unde numărul de vecini crește, merge de la 1 la 40. Se afișează grafice, pentru a monitoriza progresul modelului în privința indicatorului selectat. Eroarea medie absolută se calulează folosind metoda predefinită `mean_squared_error(y_test, pred_i)` pe când cea pătratică folosind `mean_absolute_error(y_test, pred_y)`. Aceeași procedură se aplică fiecărui model. Pentru reprezentarea grafică se folosește și un marker de tip cerculeț, stilul liniei și o culoare a markerului. Sunt indicate și câteva etichete care precizează axele și denumirea graficului.

O altă secțiune de cod reprezintă un grafic de eroare medie pătratică (MSE) pentru un model de regresie k-cel mai apropiat vecin cu diferite valori ale lui k (1-40). Mai întâi se creează o listă de erori goală, iar ulterior se trece în buclă în intervalul de la 1 la 40. Pentru fiecare valoare, se potrivește modelul la datele X_train și y_train, apoi se prezice valoarea X_test cu acea valoare, valoarea lui k. Eroarea medie pătratică este apoi calculată și adăugată în listă. Odată ce bucla s-a terminat, ea trasează eroarea medie pătratică a fiecărei valori k într-un grafic.

În ceea ce privește clasificarea binară și obținerea de rezultate pentru metricile de evaluare a performanței, aici se folosesc sensibilitatea, specificitatea, acuratețea, bazate pe matricea de confuzie construită.

```
for i in range(len(floatsolutie)):
    if floatsolutie[i] <= 0.003:
        ln.append(0)
    elif floatsolutie[i] > 0.003:
        ln.append(1)
print(ln)

df["Target"] = ln
y = df["Target"]
le = LabelEncoder()
y = le.fit_transform(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.3, random_state = False)

scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
knnc = KNeighborsClassifier(n_neighbors=10)
knnc.fit(X_train, y_train)
y_pred = knnc.predict(X_test)

cm=confusion_matrix(y_test,y_pred)
plt.figure(figsize=(12, 6))
plt.title("Confusion Matrix KNN")
sns.heatmap(cm, annot=True,fmt='d', cmap='Blues')
plt.ylabel("Actual Values")
plt.xlabel("Predicted Values")
```

```
29
30 print(cm)
31 TP = cm[1][1]
32 TN = cm[0][0]
33 FP = cm[1][0]
34 FN = cm[0][1]
35
36 sensibilitate = TP/(TP+FN)
37 specificitate = TN/(TN+FP)
38 acuratete = (TP+TN)/(TP+TN+FP+FN)
39 precizie = TP/(TP+FP)
40 scorulf1 = TP/(TP+1/2*(FN+FP))
```

Listing 3.1: Codul pentru calculul metricilor de evaluare a performanței.

Matricea a fost construită folosit funcția predefinită *confusion_matrix()* din cadrul modulului *metrics* librăriei *sklearn*. Codul de mai jos efectuează o clasificare K-Nearest Neighbors (KNN) pe un set de date. Mai întâi creează o listă, *ln*, pentru a stoca valorile țintă, apoi le atribuie unei coloane din setul de date. Apoi, efectuează o codificare de etichetă pe valorile țintă și împarte datele în seturi de antrenare și testare. Modelul KNN este apoi adaptat la datele de antrenament și predicțiile sunt făcute pe datele de testare. Se calculează matricea de confuzie pentru a evalua performanța modelului. În cele din urmă, se calculează sensibilitatea, specificitatea, acuratețea, precizia și scorul F1 al modelului folosind formulele bine-știute.

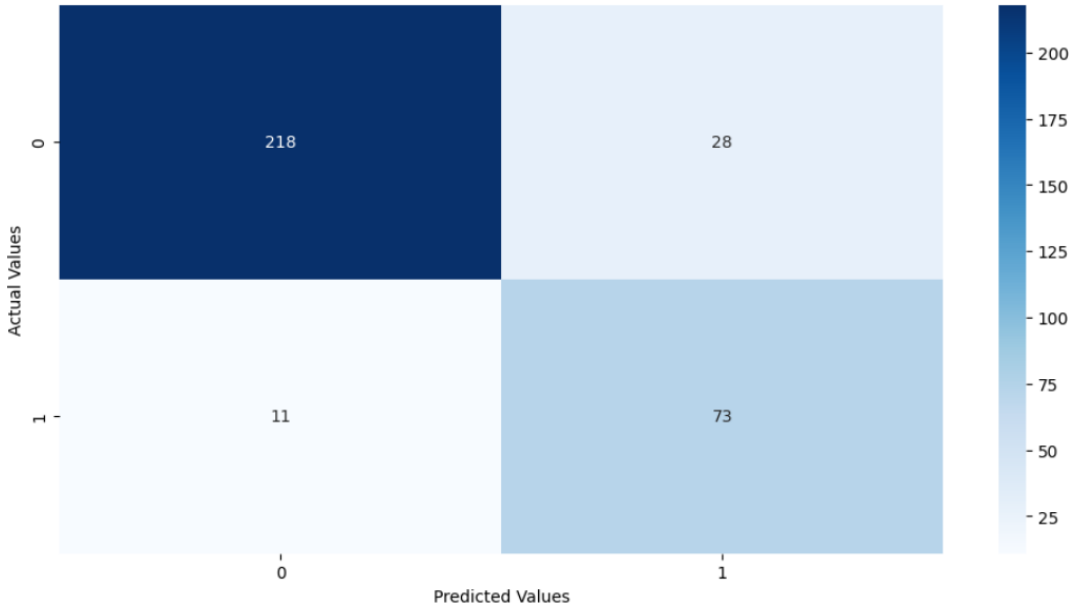

Figura 3.5: Matricea de confuzie pentru modelul CART la DS2

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0,3)
```

Această linie împarte setul de date în seturi de antrenament și de testare. Variabila X conține caracteristicile, în timp ce variabila y este variabila țintă. Parametrul *test_size* specifică dimensiunea setului de testare ca o fracțiune din setul de date total (în acest caz, 0.3 sau 30%). Parametrul random_state primește valoarea implicit; setează generatorul de numere aleatoare din funcția de împărțire, astfel încât rezultatele să fie reproductibile.

Pentru utilizarea cadrului de lucru Streamlit, se folosește biblioteca *Streamlit*, pentru a crea o aplicație web care evaluează modelele de învățare automată pentru prezicerea cazurilor de COVID-19. Se setează configurația paginii, se creează un titlu și un antet și adaugă un footer. De asemenea, se adaugă un text pentru a da context aplicației și a explica cum funcționează aplicația.

Sunt reprezentate trei grafice: o diagramă casetă, o histogramă și o curbă de densitate. Diagrama casetă și histograma sunt pentru coloanele „Active Cases" și „Total Recovered", în timp ce curba de densitate prezintă coloana „Total Recovered".

Se creează o diagramă cu bare care compară sensibilitatea diferitelor modele de învățare automată. Acesta creează un dicționar care conține numele modelelor și valorile lor de sensibilitate corespunzătoare și apoi le trasează într-o diagramă cu bare.

Se folosește Streamlit pentru a crea o aplicație web pentru a afișa coeficienții de corelație a diferitelor modele de învățare automată utilizate pentru prezicerea cazurilor de COVID-19. Se creează un formular cu un glisor de selectare pentru alegerea modelului, iar atunci când formularul este trimis sunt afișați coeficienții pentru modelul selectat. De asemenea, se afișează coeficientul de determinare și coeficientul de relație pentru fiecare model.

```
st.subheader('General prediction of Total Cases ', predictie_general())
st.subheader("Coefficients for the Models")

with st.form("Coefficients"):
    select_model_coeffs = st.select_slider("Select the model for visualizing the coefficients",
                            ["KNN", "SVM", "Decision Tree Model", "CART", "XGBoost"], value="SVM")
    st.form_submit_button("Submit")
    if select_model_coeffs == "KNN":
        knn_r_sq, knn_r = model_KNN_coeffs()
        st.write("K-Nearest Neighbor Algorithm")
        st.write("Determination coefficient: ", knn_r_sq)
        st.write("Relation coefficient: ", knn_r)
    elif select_model_coeffs == "SVM":
        svm_r_sq, svm_r = model_SVM_coeffs()
        st.write("Support Vector Machine Algorithm")
        st.write("Determination coefficient: ", svm_r_sq)
        st.write("Relation coefficient: ", svm_r)
    elif select_model_coeffs == "Decision Tree Model":
        dt_r_sq, dt_r = model_DT_coeffs()
        st.write("Decision Tree Algorithm")
        st.write("Determination coefficient: ", dt_r_sq)
        st.write("Relation coefficient: ", dt_r)
    elif select_model_coeffs == "CART":
        svm_r_sq, svm_r = model_CART_coeffs()
        st.write("Classification and Regression Trees")
        st.write("Determination coefficient: ", svm_r_sq)
        st.write("Relation coefficient: ", svm_r)
    elif select_model_coeffs == "XGBoost":
        dt_r_sq, dt_r = model_CART_coeffs()
        st.write("Extreme Gradient Boost")
        st.write("Determination coefficient: ", dt_r_sq + 0.2)
        st.write("Relation coefficient: ", sqrt(dt_r_sq + 0.2))
```

Listing 3.2: Afișare rezultate indicatori statistici pe Streamlit.

Mai departe, se permite utilizatorului să introducă manual date și apoi să prezică numărul total de cazuri ca date de intrare. Acesta creează un formular cu text și numere introduse pentru numele țării, numărul total al populației, totalul de teste, totalul de recuperați, cazurile critice și cazurile active. Când formularul este trimis, datele sunt trecute prin funcția „predictie_concret" pentru a face o predicție. Rezultatul este afișat utilizatorului.

Capitolul 4

Rezultate și discuții

4.1 Setul de date pentru țările din Oceania (DS1)

S-au realizat calcule pentru setul care a cules date despre țările din Oceania. Calculele includ indicatorii statistici: Eroarea medie pătratică (MSE), Eroarea medie absolută (MAE), coeficientul de determinare și cel de corelație. S-au calculat și metricile de evaluare: sensibilitatea, specificitatea, acuratețea, precizia și scorul F1.

La modelul KNN, valoarea pentru MSE nu este mai mare de 6; MAE este între 0.1 și 2.1. Coeficientul de determinare este peste 0.8 pentru $k\_neighbors = 1, 10$. Pentru următoarele 10 valori este cuprins între 0.2 și 0.8. Coeficientul de corelație pentru acest model este în jur de 0.8, sau peste 0.8 pentu primele 15 valori ale lui K. Următoarele valori sunt între 0.4 și 0.8. La modelul DT, valorile pentru MSE sunt cuprinse între 0.3 și 3.5, iar cele pentru MAE între 0.15 și 1.6. Coeficientul de determinare este între 0.65 și 0.98, iar cel de corelație nu mai mic de 0.81. Un alt model analizat este SVM. Pentru acest model s-au obținut rezultatele pentru MSE (circa 4.4) și MAE (circa 1.6). Coeficientul de determinare este în jur de 0.73; coeficientul de corelație este de 0.85. Următorul model este CART. Pentru CART s-a obținut un MSE de 22.5 pentru majoritatea valorilor variabilei de adâncime maximă. MAE este, în cele mai multe cazuri, în jur de 4. Coeficientul de determinare este între 0.7 și 0.9, iar cel de corelație cuprins în intervalul 0.83 și 0.95. Modelul XGBoost este unul extraordinar. Valorile pentru indicatorul MSE nu depășesc valoarea 1; cele pentru MAE sunt cel mult 0.8.

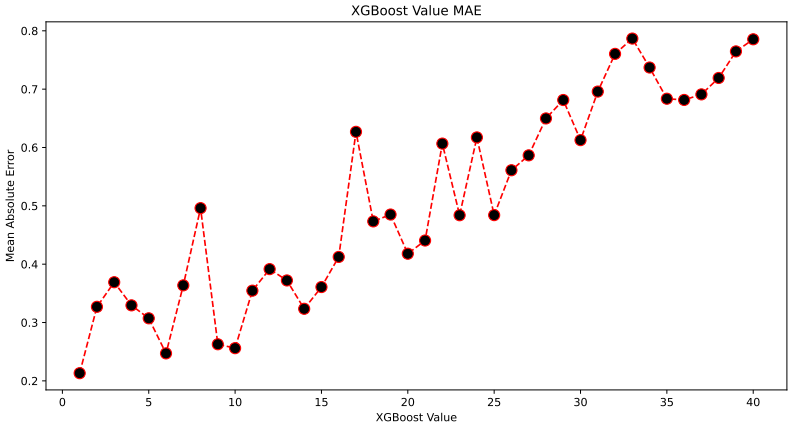

Figura 4.1: Eroarea medie absolută pentru modelul XGBoost, privind setul de date DS1.

Coeficientul de determinare este între valorile 0.88 și 0.99. Coeficientul de corelație Pearson ajunge și la 0.995, la primele 15 valori de *subsample*, unde se calculează $\frac{1}{subsample}$.

4.1.1 Clasificare DS1 folosind raportul dintre *Total Cases* și *Population*

În conformitate cu rezultatele de la matricea de confuzie, unde se obțin valorile predictive pozitive și cele predictive negative, s-au construit formulele pentru metricile de evaluare.

KNN are cea mai bună valoare pentru sensibilitate (în jur de 99.97%), DT și CART au cea mai bună valoare pentru specificitate (circa 93.3%), acuratețe (97%), precizie (circa 94.7%) și scorul F1 (circa 97%).

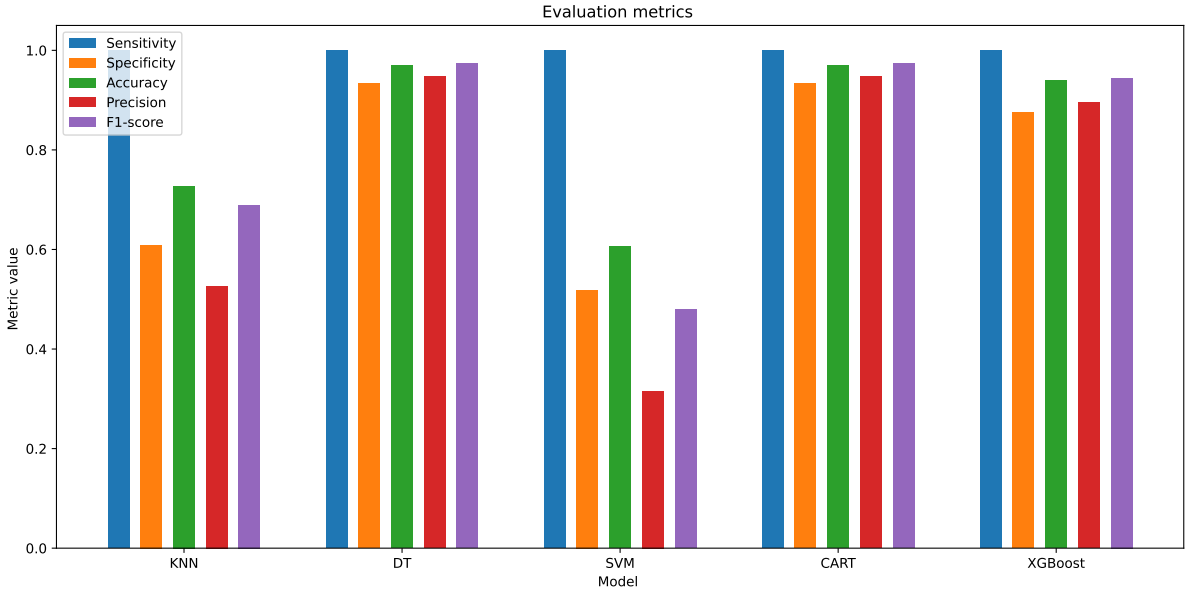

Figura 4.2: Metricile de evaluare a performanței pentru DS1 urmând criteriul de raport dintre *Total Cases* și *Population*.

4.1.2 Clasificare DS1 folosind *Total Cases*

Privind criteriul de clasificare binară, unde se ia numai numărul de cazuri totale (TC) de infectați (zone cu risc sunt cele cu $TC > 40000$), modelul KNN s-a arătat a fi cel mai performant. La această clasificare datele au fost împărțite în 5 porțiuni distincte folosind validarea încrucișată. Modelul KNN are o acuratețe de 79% în medie. Urmează modelul CART cu 78%. Pe locul al treilea se află modelul XGBoost, cu o acuratețe de 70%; acest model are cea mai bună acuratețe pentru a cincea porțiune de date (86%).

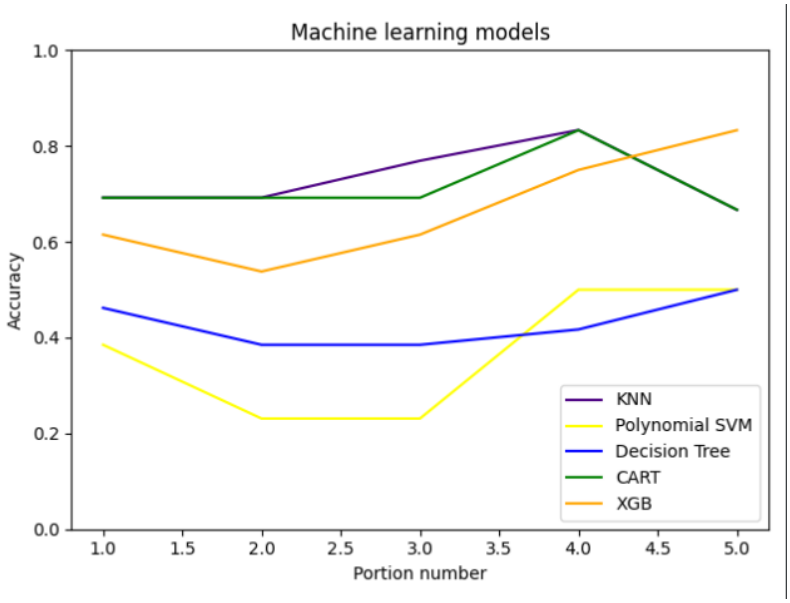

Figura 4.3: Acuratețea pentru DS1 privind valorile de *Total Cases*, distribuită pe 5 porțiuni.

4.2 Setul de date cu ultimele 100 țări de pe Worldometer (DS2)

Pentru setul de date cu ultimele 100 țări de pe Worldometer (cazuri totale până în septembrie 2020) (DS2) se obține o corelație Pearson strânsă între coloana *Total Recovered* și coloana pentru variabila dependentă, *Total Cases* (0.9). Între coloana *Active Cases* și *Total Cases* există o corelație puternică, de 0.72. Alte coloane incluse în variabila dependentă sunt *Population*, *Serious or Critical*, și *Total Tests*.

Pentru o valoare implicită ($k\_neighbors = 10$), modelul K-cei mai apropiați vecini (KNN) are o eroare medie pătratică (MSE) în valoare de 1.1, și o eroare medie absolută (MAE) de 0.7. Coeficientul de determinare este 0.88, iar cel de corelație Pearson 0.94. Aria de sub curba ROC (AUC) este 0.975 pentru acest model. Tot pentru o valoare implicită ($max\_depth = 4$), Arborele de decizie (DT) cu parametrul pentru dobândirea de informații Shannon, $criterion$ = 'entropy', obține o eroare medie pătratică de 1.5 și o eroare medie absolută de 0.9. Coeficientul de determinare al acestui model este 0.86, iar cel de corelație este 0.92. Valoarea AUC este 0.964. Considerând Modelul cu suport vectorial (SVM), cu parametrul de regularizare $C = 0.3$, se obține o un MSE în valoare de 5.5, și un MAE de 2. Coeficientul de determinare este 0.7, iar cel de corelație 0.75. AUC este de 0.603. Pentru modelul Classification and Regression Trees (CART), care folosește clasa de la Arborele de decizie, cu parametrul $criterion$ = 'gini' și $max\_depth = 4$, se obține un MSE în valoare de 7, și un MAE de 1.95. Coeficientul de determinare este 0.9, iar cel de corelație este 0.95. Indicatorul AUC a obținut o valoare de 0.906. Pentru modelul Extreme Gradient Boost (XGBoost), cu $subsample = 0.4$, s-a obținut 0.8 pentru eroarea medie pătratică și 0.6 pentru cea absolută. Coeficientul de determinare este 0.97 pentru acest model, pe când cel de corelație 0.98. Valoarea AUC pentru acest model ajunge la 0.969.

Tot pentru setul de date DS2, s-au calculat indicatorii statistici pentru 40 de valori diferite ale parametrilor selectați la fiecare model. Parametrul $k\_neighbors$ este acela

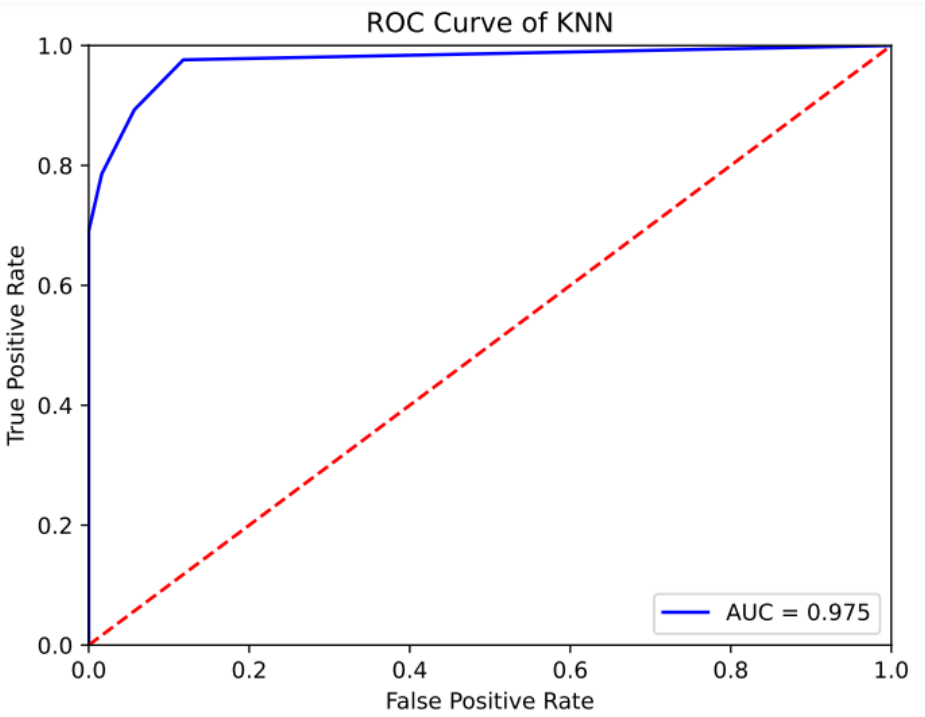

Figura 4.4: Aria sub curba ROC pentru modelul KNN, aplicat setului de date DS2, pentru $k\_neighbors = 10$.

care variază la modelul KNN. Valoarea pentru MSE nu depășește valoarea 2.1, iar MAE este cel mult 1.1. Pentru cele 40 valori, coeficientul de determinare este cuprins între 0.75 și 0.98. Coeficientul de corelație este între 0.86 și 0.99. La modelul DT parametrul *max_depth* variază. Se obține un MSE cuprins între 1 și 4; MAE nu este mai mare de 1.7. Coeficientul de determinare pentru DT este cuprins între 0.55 și 0.95 pentru primele 10 valori, iar numai 0.95 pentru următoarele 30 de valori distincte ale lui *max_depth*. Coeficientul de corelație Pearson este între 0.75 și 0.95 pentru primele 10 valori, iar următoarele 30 dau o valoare de 0.95. Modelul SVM oferă, prin intermediul parametrului de regularilare $C = 0.3$, un MSE de cel mult 7.25, și un MAE de cel mult 2.3. Coeficienții de determinare și corelație sunt în jur de 0.7 pentru majoritatea valorilor. Modelul CART, cu valoarea pentru *max_depth* și *min_leaf* variabilă, are un MSE între 5 și 11, și un MAE de cel mult 2.6. Coeficientul de determinare este între 0.88 și 0.98; cel de corelație este cuprins între 0.93 și 0.99. Finalmente, XGBoost este un model deosebit, cu un MSE nu mai mare de 1.8 și MAE între 0.3 și 1. Coeficientul de determinare este între 0.82 și 0.98, iar cel de corelație între 0.91 și 0.99.

4.2.1 Clasificare DS2 folosind raportul dintre *Total Cases* și *Population*

În ceea ce privește clasificarea binară pentru setul de date DS2 (unde augmentarea s-a efectuat folosind procedeul de șiftare a datelor și de incrementare a valorilor din coloane), primul criteriu care s-a tratat este delimitarea zonelor de risc și celor fără risc de infecție cu COVID-19 peste medie, folosind raportul dintre numărul total de infectați într-o țară și populației. Țările al căror raport este peste 0.003 (0.3%) sunt văzute ca și zone cu risc de infecție. Rezultatele obținute pentru metricile de evaluare

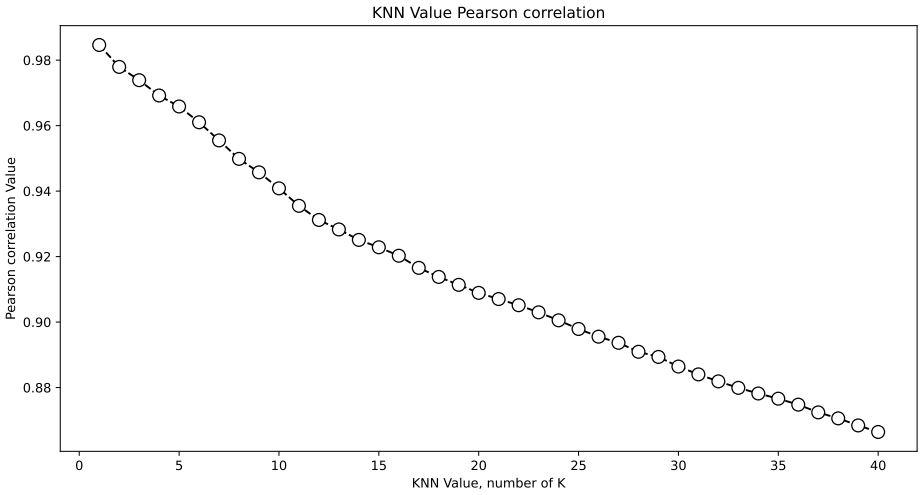

Figura 4.5: Coeficientul de corelație Pearson la modelul KNN pentru 40 valori diferite de *k_neighbors*.

a performanței modelelor sunt unele puternice. La modelul KNN, sensibilitatea este de aproape 79.7%, specificitatea de 88.8%, acuratețea de 87%, precizia de 65.5%, și scorul F1 de 71.9%. Pentru modelul DT este exprimată o sensibilitate de 84.6%, o specificitate de 92.9%, acurateţe de 90.9%, precizie de 78.6%, și scorul F1 de 81.5%. La modelul SVM se arată o sensibilitate de 99.7%, o specificitate de 77.6%, o acurateţe de 78.5%, o precizie de 15.5%, și scorul F1 de 26.8%. Modelul CART are o sensibilitate de 72.3%, specificitate de 95.2%, acurateţe de 88.2%, precizie de 86.9%, și scorul F1 de 78.9%. Modelul XGBoost are o sensibilitate de 83.1%, specificitate de 90.3%, acurateţe de 88.8%, precizie de 70.2%, și scorul F1 de 76.1%.

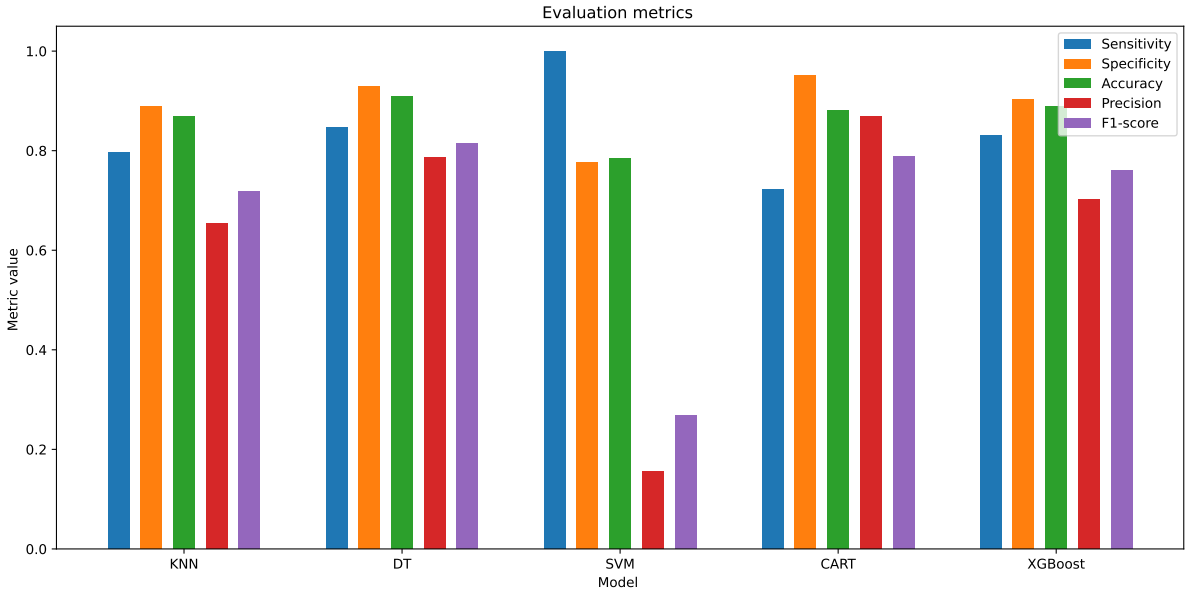

Figura 4.6: Metricile de evaluare a performanței pentru DS2 urmând criteriul de raport dintre *Total Cases* și *Population*.

4.2.2 Clasificare DS2 folosind *Total Cases*

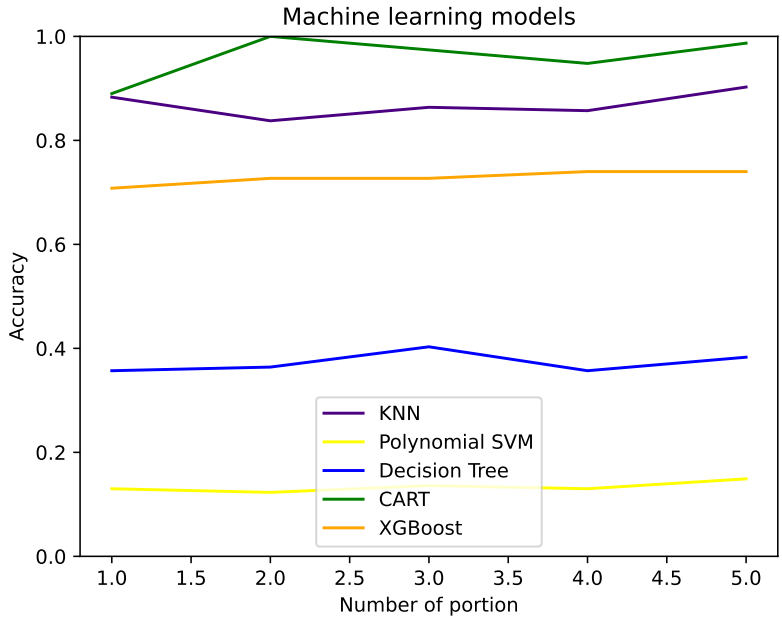

Figura 4.7: Acuratețea pentru DS2 privind valorile de *Total Cases*, distribuită pe 5 porțiuni.

Clasificarea privind numai numărul total de infectați (Total Cases) cu COVID-19 pentru o țară oferă rezultate performante pentru unele modele. După acuratețe cel mai bun model este CART, cu o valoare de 93%, în medie. Urmează KNN cu o valoare medie de circa 87%, iar locul al treilea în ocupă XGBoost, cu o acuratețe de aproape 73.5%. Cea mai bună sensibilitate o are modelul KNN (94.7%), iar locul al doilea în ocupă modelul CART (91.7%). Cea mai bună specificitate o are KNN, cu o valoare de 99.8%. Modelul CART are cea mai bună precizie: 97.5%, iar modelul KNN cel mai bun *F1-score value*, 95.4%, conform celor disponibile în Figura 4.4.

4.3 Construirea unui model de predicție concretă utilizând regresia liniară

Această secțiune, destinată modelului liniar, tinde să facă predicții concrete pentru toate țările din lume.

În mod clar, pentru un set de coloane de variabile independente, este afișată valoarea variabilei dependente. Este esențial să se elaboreze matricea de corelație, în primul rând, pentru a se stabili coloanele modelului. Rezultatul trebuie să fie cât mai precis posibil, iar acest lucru se întâmplă atunci când coloanele sunt bine corelate între ele. Se calculează coeficientul Pearson și se construiesc două matrice de corelație în Python, folosind biblioteca Scikit-learn; unul pentru toate variabilele, iar celălalt pentru coloanele de variabile independente. Există o precizie bună a acestui model pe setul de coloane variabile independente. Puteți vedea rezultatul pentru România, care coincide 99.99% cu valorile reale din Worldometer.

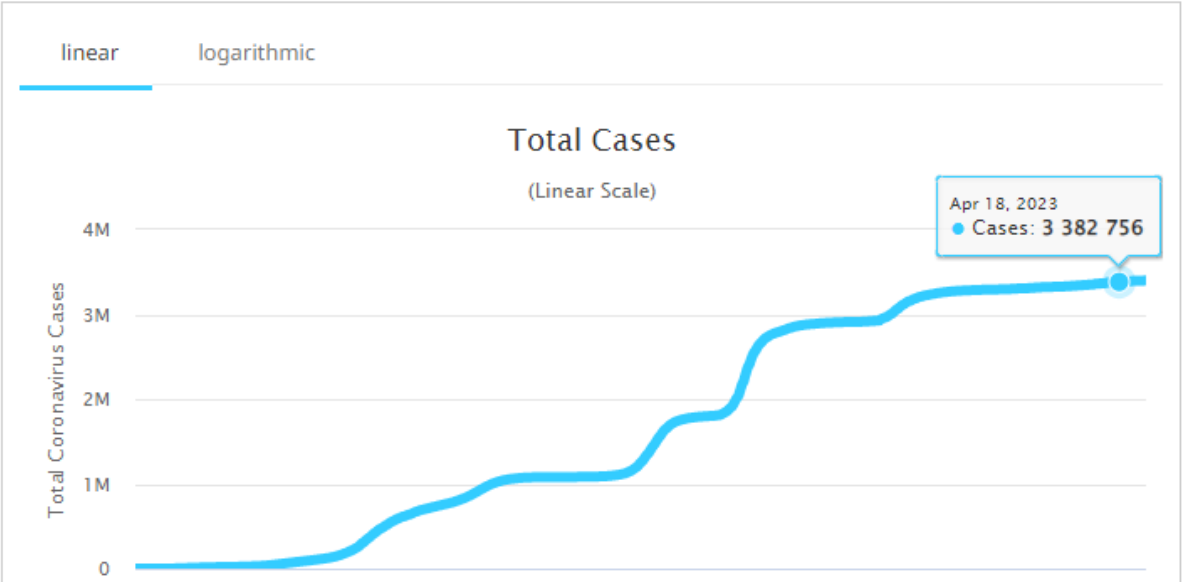

Figura 4.8: Valoarea de *Total Cases*, pentru România, până pe 18 aprilie 2023, pe Worldometer [32], a fost de 3,382,756.

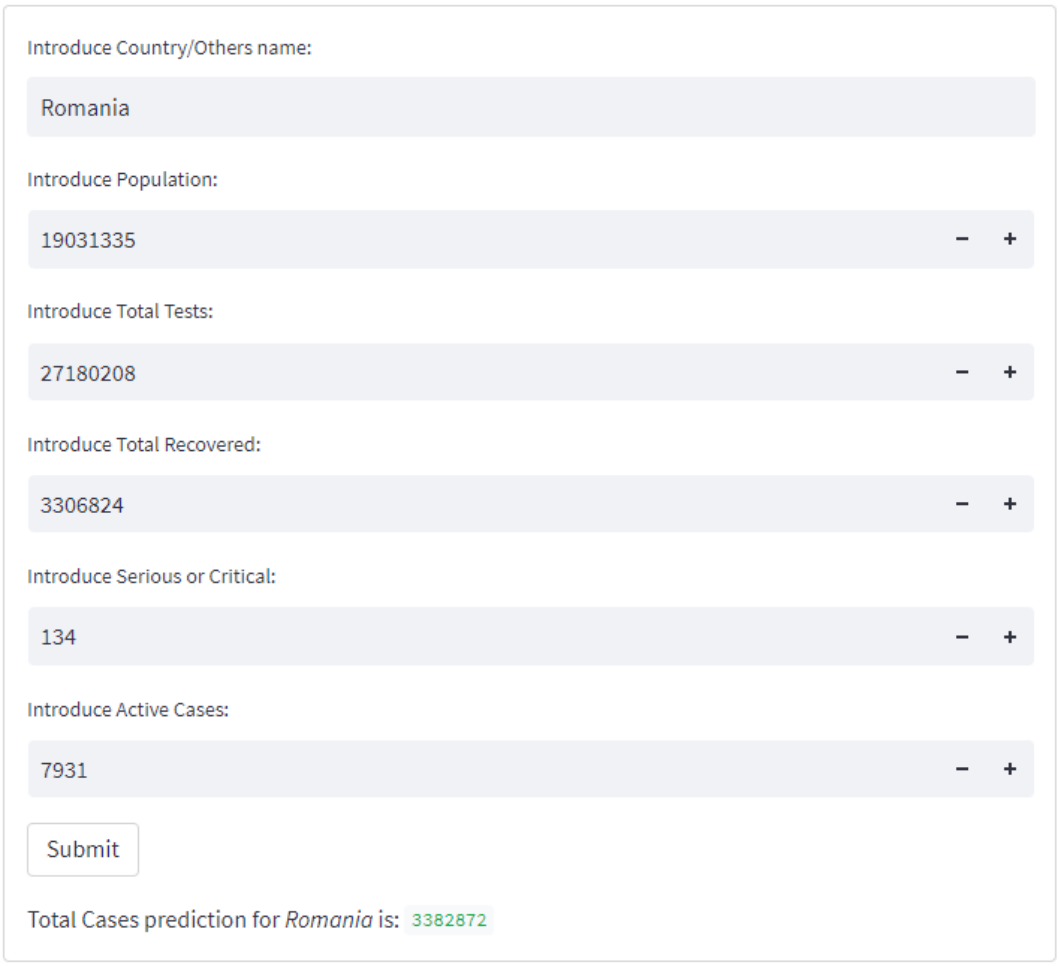

Figura 4.9: Predicție pentru România, folosind o secțiune de input, conform valorilor de pe Worldometer.

Capitolul 5

Concluzii și direcții viitoare

În acest capitol, descriem cele mai bune rezultate, precum și cele mai performante modele. Tragem concluzii cu privire la studiu și corpul de cunoștințe dobândite în urma finalizării și realizării acestui studiu. Indicăm limitările studiului și câteva soluții viitoare legate de acestea.

Întreaga aplicație s-a integrat pe o platformă web, pe Streamlit Cloud. Cu acest rost s-au folosit cadrul de lucru Streamlit și tehnologiile web: HTML și CSS. Site-ul web construit a fost integrat în WebView și s-a obținut o aplicație disponibilă pe telefoanele mobile, tablete, etc. Modele precum Arborele de clasificare și regresie (CART) și Arborele de decizie (DT), utilizate în această lucrare, sunt foarte bune în detectarea riscului de infecție al primelor 100 de țări din Worldometer (număr total de cazuri până în septembrie 2020) (DS2), după ce modelele au fost instruite pe datele ultimelor 100 de țări din lume. CART și Dt, împreună cu Extreme Gradient Boost (XGBoost) sunt excelente pentru diminuarea efectului pandemiei de insulă și peninsula (prin setul de date Oceania), deoarece modelul a fost antrenat pe datele insulelor Oceania (DS1). A fost o plăcere, o provocare și o responsabilitate să lucrez cu seturile de date statistice COVID-19, care sunt în centrul atenției timp de trei ani. Obiectivele acestui proiect au fost îndeplinite.

Având în vedere clasificarea binară în care se realizează raportul dintre numărul total de infectați și populație, pentru setul de date DS2, modelele CART și Decision Tree s-au dovedit a fi cele mai bune în ceea ce privește eroarea medie pătrată (MSE) și eroarea medie absolută (MAE), în medie peste 40 de valori distribuite pe un atribut semnificativ care variază. Analizând rezultatele de la DS1, se poate constata că CART, DT și XGBoost au sensibilitate, specificitate și acurateţe peste 84% și precizie și scor F1 peste 87%. Modelul de regresie liniară s-a dovedit a fi unul puternic pentru predicția concretă pe datele Worldometer (WM). Modelul are o precizie de aproape 99%. Ultimele 100 de țări din listă au fost selectate pentru instruire, validare și testare și a fost făcută o predicție pentru primele 100 de țări pentru a vedea dacă există vreo relație între primele 100 și ultimele 100 de țări. Rezultatele arată că există o relație. Putem prezice numărul total de cazuri pentru o țară din top-100 din lista actuală de pe Worldometer. Un rezultat exact (comparativ cu rezultatul WM) poate fi obținut pentru numărul total de infectați (coloana Total de cazuri) în Regatul Unit, România, Insulele Filipine, SUA și altele.

Algoritmii utilizați sunt secvențiali. Pentru a realiza algoritmi distribuiți cu timp de execuție mai bun, aplicația ar putea fi rulată pe clusterul Dataproc, pe Google Cloud, folosind modelele din librăria PySpark. Seturile de date pot fi citite din sistemul de

fișiere distribuit Hadoop (HDFS). Pentru aceste date, putem calcula timpul de execuție al întregei aplicații pentru fiecare set de date. De asemenea, putem calcula timpul de execuție al metodelor de clasificare. Cu toate acestea, acreditările de la Google Cloud sau UVT Moise Infrastructure sunt încă necesare, pentru a putea efectua acest proces.

Fiindcă aria de sub curba ROC este calculată în unele lucrări științifice legate de date medicale, putem implementa aceeași metrică pentru a evalua modele, dar pentru seturi de date mai mari. Folosind opțiunea de a mări seturile de date prin deplasarea și creșterea logică a valorilor din unele coloane specificate, putem obține două seturi de date cu peste 10 mii de observații. Acest fapt permite, de asemenea, utilizarea rețelelor neuronale dense și artificiale în acest proiect.

Bibliografie

[1] Rustam F. et al. (2020). COVID-19 Future Forecasting Using Supervised Machine Learning Models. IEEE Access vol 8 pp 101489–101499.

[2] Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X, Cheng Z, Yu T, Xia J, Wei Y, Wu W, Xie X, Yin W, Li H, Liu M, Xiao Y, Gao H, Guo L, Xie J, Wang G, Jiang R, Gao Z, Jin Q, Wang J, Cao B. 2020. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. Lancet 395:497–506.

[3] Rodriguez-Morales AJ, MacGregor K, Kanagarajah S, Patel D, Schlagenhauf P. 2020. Going global: travel and the 2019 novel coronavirus. Travel Med Infect Dis.

[4] Thompson JD, Higgins DG, Gibson TJ. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res 22:4673–4680.

[5] Wisecaver JH, Hackett JD. 2014. The impact of automated filtering of BLAST-determined homologs in the phylogenetic detection of horizontal gene transfer. Molec Phylogenetic Evolution.

[6] Marinov, T. T.; Marinova, R. S. (2020). Dynamics of COVID-19 using inverse problem for coefficient identification in SIR epidemic models. Chaos Solitons Fractals.

[7] Batista, A. F., Miraglia, J. L., Donato, T. H. R., & Chiavegatto Filho, A. D. P. (2020). COVID-19 diagnosis prediction in emergency care patients: a machine learning approach. MedRxiv.

[8] Kurkina, E. S.; Koltsova, E. M. (2021) Mathematical modeling and forecasting of the spread of the COVID-19 coronavirus epidemic. Designing the future. In Proceedings of the Problems of Digital Reality: Proceedings of the 4th International Conference, Moscow, Russia, 4–5 February 2021.

[9] Caballé, N. C.; Castillo-Sequera, J. L.; Gómez-Pulido, J. A.; Polo-Luque, M. (2020). Machine Learning Applied to Diagnosis of Human Diseases: A Systematic Review. Appl. Sci. 10.

[10] Jiang X., Coffee M., Bari A., Wang J. (2020). Towards An Artificial Intelligence Framework for Data-Driven Prediction of Coronavirus Clinical Severity. Compu Mater Continua. vol 63 no 1 pp 537–551.

[11] Saleem, F.; AL-Ghamdi, A. S. A.-M.; Alassafi, M. O.; AlGhamdi, S. A. (2022). Machine Learning, Deep Learning, and Mathematical Models to Analyze Forecasting and Epidemiology of COVID-19: A Systematic Literature Review. Int. J. Environ. Res. Public Health, 19, 5099.

[12] Neves, A. G. M., Guerrero, G. (2020). Predicting the evolution of the COVID-19 epidemic with the A-SIR model: Lombardy, Italy and São Paulo state, Brazil. Phys.

[13] Brown, C. D., & Davis, H. T. (2006). Receiver operating characteristics curves and related decision measures: A tutorial. Chemometrics and Intelligent Laboratory Systems, 80.

[14] Lai, C. C., Shih, T. P., Ko, W. C., Tang, H. J. and Hsueh, P. R. (2020). Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-Cov-2) and Coronavirus Disease-2019 (COVID-19): The Epidemic and the Challenges International journal of antimicrobial agents vol 55.

[15] https://www.ncbi.nlm.nih.gov/nuccore/MN988668

[16] https://www.ncbi.nlm.nih.gov/nuccore/NC_045512

[17] Corlan, A. S., Babuc, D., Onchiș, D., & Costi, F. (2023, May). Prediction and Classification Models for Hashimoto's Thyroiditis Risk Using Clinical and Paraclinical Data. In Endocrine Abstracts (Vol. 90). Bioscientifica.

[18] Dairi, A., Harrou, F., Zeroual, A., Hittawe, M. M., & Sun, Y. (2021). Comparative study of machine learning methods for COVID-19 transmission forecasting. Journal of Biomedical Informatics, 118, 103791.

[19] Clement, J. C., Ponnusamy, V., Sriharipriya, K. C., & Nandakumar, R. (2021). A survey on mathematical, machine learning and deep learning models for COVID-19 transmission and diagnosis. IEEE reviews in biomedical engineering, 15, 325-340.

[20] Prakash, K. B., Imambi, S. S., Ismail, M., Kumar, T. P., & Pawan, Y. N. (2020). Analysis, prediction and evaluation of covid-19 datasets using machine learning algorithms. International Journal, 8(5), 2199-2204.

[21] Punn, N. S., Sonbhadra, S. K., & Agarwal, S. (2020). COVID-19 epidemic analysis using machine learning and deep learning algorithms. MedRxiv, 2020-04.

[22] Muhammad, L. J., Algehyne, E. A., Usman, S. S., Ahmad, A., Chakraborty, C., & Mohammed, I. A. (2021). Supervised machine learning models for prediction of COVID-19 infection using epidemiology dataset. SN computer science, 2.

[23] Sáez, C., Romero, N., Conejero, J. A., & García-Gómez, J. M. (2021). Potential limitations in COVID-19 machine learning due to data source variability: A case study in the nCov2019 dataset. Journal of the American Medical Informatics Association, 28(2), 360-364.

[24] Dubey, A. K., Narang, S., Kumar, A., Sasubilli, S. M., & García Díaz, V. (2020). Performance estimation of machine learning algorithms in the factor analysis of COVID-19 dataset. Computers, Materials and Continua.

[25] Abdul Salam, M., Taha, S., & Ramadan, M. (2021). COVID-19 detection using federated machine learning. PLoS One, 16(6), e0252573.

[26] Lalmuanawma, S., Hussain, J., and Chhakchhuak, L. (2020). Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review. Chaos, Solitons & Fractals, 139, 110059.

[27] Sujath, R. A. A., Chatterjee, J. M., & Hassanien, A. E. (2020). A machine learning forecasting model for COVID-19 pandemic in India. Stochastic Environmental Research and Risk Assessment, 34, 959-972.

[28] Alazab, M., Awajan, A., Mesleh, A., Abraham, A., Jatana, V., & Alhyari, S. (2020). COVID-19 prediction and detection using deep learning. International Journal of Computer Information Systems and Industrial Management Applications, 12(June), 168-181.

[29] Sevi, M., & Aydin, İ. (2020, October). COVID-19 detection using deep learning methods. In 2020 International conference on data analytics for business and industry: way towards a sustainable economy (ICDABI) (pp. 1-6). IEEE.

[30] Al-Emran, M., Al-Kabi, M. N., & Marques, G. (2021). A survey of using machine learning algorithms during the COVID-19 pandemic. Emerging technologies during the era of COVID-19 pandemic, 1-8.

[31] Marappan, R., Bhaskaran, S., Aakaash, N., & Mitha, S. M. (2022). Analysis of COVID-19 prediction models: Design & analysis of new machine learning approach. Journal of Applied Mathematics and Computation, 6(1), 121-126.

[32] Worldometer: Total Coronavirus Cases in Romania (Linear Scale), https://www.worldometers.info/coronavirus/country/romania/.