Proiect Învățare automată Alergare

Mihai Crisan

May 2024

1 Introducere

Proiectul are ca scop dezvoltarea și antrenarea unui sistem inteligent capabil de a recunoaște dacă o persoană fuge sau merge normal in funcție de datele obțiunte de 2 senzori (gyroscop și accelerometru) aflați în telefonul unei persoane. Motivația alegerii proiectului și a setului de date este începerea regulată de a exersa, în principal prin alergare.

Mersul și alergarea sunt cea mai practicată respectiv a doua cea mai practicată activitate în lume, fiind parte din viețile noastre de zi cu zi.

Acest sistem inteligent este gândit cu scopul utilitar de a detecta în timp real dacă o persoană este angajată în alergare, astfel eliminând nevoia de a porni un program de fitness manual și fără a mai avea grija că o perioadă de fugă nu a fost înregistrată.

Dacă sistemul inteligent prezintă o rată de succes ridicată și integrarea este realizată cu succes, aplicațiile mobile ar porni modurile de antrenament aferente singure, eliminând perioada inconfortabilă de a porni un antrenament.

Tehnologii folosite: Python: limbajul de programare folosit în dezvoltarea proiectului, Jupyter: un tip de formatare a documentelor dezvoltate pe baza JSON, Pandas: o librărie folosită în manipularea seturilor mari de date, XGBoost: o librărie folosită pentru implementarea

2 Context

2.1 Context general

Baza de date a fost aleasă de pe Kaggle, datele fiind colectate de utilizatorul Viktor Malyi utilizând un telefon iPhone 5c. În tabelă sunt înregistrate 88588 de valori ale dăți, a timpului la care au fost înregistrate alături de valoarea fiecărei senzor.

Senzorii folosiți în colectarea datelor sunt accelerometrul și gyroscopul integrat în telefon. Din cei doi senzori sunt înregistrate câte 3 valori pe axele inălțimii, lățimii și a lungimii,pentru accelerometru si pe 3 axe pentru gyroscop. Datele înregistrate de senzori sunt numere reale, cu valori cuprinse între -5.35 și 5.6 pentru accelerometru și intre -7.46 si 8.5 pentru gyroscop.

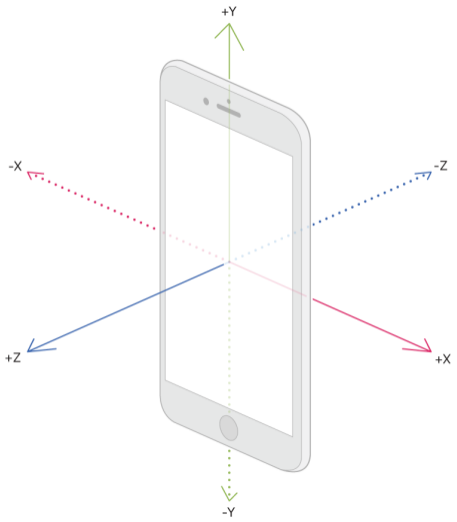

Figure 1: Axele senzorilor prezenți în telefon

În Figura 1 sunt prezentate axele de colecție a datelor. Este important de menționat că datele au o variație mare deoarece la colectarea lor rotația telefonului are un rol important în determinarea semnului datelor. Acest lucru oferă o acuratețe mai mare a algoritmilor antrenați pe aceste seturi de date și permit implementarea modelui în o gamă mai largă de aplicații.

2.2 Tema Proiectului

Tema proiectului este de a dezvolta un sistem de învățare automată folosind date reale și accesibile oricui prin intermediul internetului. Proiectul este împărțit în 7 etape principale, fiecare fiind o parte importantă din dezvoltarea sistemului.

2.3 Rezultate Propuse

La finalul proiectului dorim să obținem un sistem inteligent capabil de a detecta dacă o persoană se află în fugă sau în mers pe baza input-urilor senzorilor înregistrate în setul de date. Acuratețea propusă pe baza datelor este de peste 70%.

3 Aspecte teoretice

3.1 Starea domeniului

Domeniul inteligenței artificiale și a sistemelor în starea lor curentă este unul vast. De la antrenarea sistemelor în a detecta celule canceroase, la strategiști

politici ce folosesc sisteme de predicție pentru a calcula următoarele mișcări electorale, aspecte evidențiate de Min Yin et al. [1], sistemele inteligente devin din ce în ce mai ușor de incorporat în orice aplicație ce poate beneficia de analiza seturilor mari de date sau de predicții cu o acuratețe mare. Utilizarea sistemelor inteligente implică o bună cunoștință a multor aspecte teoretice relevante. În următoarele subcapitole sunt descrise principalele concepte folosite în realizarea acestui proiect.

3.2 Entropia

În domeniul inteligenței artificiale entropia poate fi definită ca fiind o unitate de masură pentru a determina nivelul de dezordine sau de incertitudine dintr-un sistem. Este folosită în a determina calitatea unui set de date, indicând dacă setul respectiv conține informații distribuite. Presupunem următoarele cazuri:

Figure 2: Entropie Minimă

Figure 3: Entropie Ridicată

Din figurile prezentate mai sus se observă că în Figura 2 avem un set de date cu entropie mică (entropia este chiar 0, deoarce toate punctele sunt identice) iar in Figura 3 avem un set de date cu entropie relativ ridicată.

Intr-un set de date cu C clase entropia este calculată utilizând formula **??**:

$$E = -\sum_{i}^{C} *p_i * log_2(p_i) \tag{1}$$

Unde pi este probabilitatea de a alege un element din mulțimea setului de date (i.e. proporția setului de date care este i)

Originea entropiei în domeniul științei calculatoarelor și a teoriei numerelor se găsește în lucrarea scrisă de C.E. Shannon [2], termenul fiind împrumutat din domeniul termodinamicii, referindu-se la dezordinea unui sistem.

3.3 Corelații

Corelațiile dintre date ne arată relevanța unor anumite clase de date față de alte clase de aparținând aceluiaș set de date. Calcularea corelațiilor intr-un set de date ne ajută în a dezvolta sisteme ce au acuratețe mai mare, concept explicat de Mark A. Hall [3], explicând cum în domeniul sistemelor inteligente se pot perfecționa sistemele utilizând caracteristicile selectate pe baza corelațiilor identificate între ele.

În lumea reală corelațiile nu promit că două evenimente sunt în directă legătură una față de cealaltă, insă pot indica inspre direcția potrivită. Evenimente precum detectarea realității atacurilor cibernetice, explorate de Ibraham Ghafir et al. [4], sau identificarea unei stări emoționale a unei persoane, indicate de Sunil Kumar și Ilyoung Chong [5], nu pot fi prezise în acuratețe maximă, însă prin analizarea datelor se pot obține linii principale care indică spre un anumit rezultat.

3.4 Modele

În proiect au fost alese inițial 4 tipuri de modele pentru a fi testate pentru implementarea finală: Regresia Liniară, Regresia Logistică, LSTM și XGBoost. În continuare sunt descrise fiecare model și o descriere simplificată a fiecărui tip de algoritm.

3.4.1 Regresie liniară

Regresia liniară este unul dintre algoritmii de bază în dezvoltarea sistemelor inteligente. Regresia este folosită în a determina corelațiile dintre două sau mai multe variabile având relații de cauză-efect și pentru a oferi predicții folosind relațiile dintre obiecte. Acest lucru îl considerăm cunoștință generală, însă definiția oferită de Gülden KAYA UYANIK și Neşe Güler în articolul "A study on multiple linear regression analysis" [6] oferă o perspectivă generală foarte bună asupra modelului.

3.4.2 Regresia Logistică

Regresia logistică este asemănătoare în funcționalitate cu regresia liniară, singura diferență fiind modul de determinare a valorilor maxime.

Regresia logistică are valorile "blocate" intre o valoare minimă și o valoare maximă, spre deosebire de regresia liniară ce explunele datele în continuitate

În Figura 4 se observă diferența dintre distribuirea datelor în o regresie liniară față de o regresie logistică, mai exact pragurile în care datele sunt considerate o valoare prestabilită.

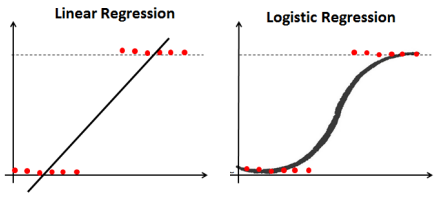

Figure 4: Diferența regresiilor

3.4.3 Long Short Term Memory

LSTM se încadrează în clasa sistemelor recurente, folosind reprezentări a inputuri recent introduse (short term), față de parametrii modificați în timp (long term). Sepp Hochreiter și Jurgen Schimdhuber explică cum acest sistem poate învăța să creeze legături între informații introducând o eroare constantă în reținerea informațiilor [7].

Aceste sisteme se bazează pe truncarea în diferite serii a datelor, în proiect ele fiind împărțite în o serie temporală.

Acest algoritm nu a prezentat rezultate satisfăcătoare din perspectiva acurateței, astfel nu a fost dezvoltat în detaliu.

3.4.4 XGBoost

XGBoost (eXtrem Gradient Boosting) face parte din categoria "Gradient Boosting", ce s-a dovedit în timp ca fiind una dintre cele mai rapide și acurate sisteme inteligente. Conform lui Didrik Nielsen [8] acest algrotim oferă "rezultate remarcabile pentru o varietate mare de probleme".

Algoritmii de tip gradient boost încearcă să gasească o funcție a căror rezultate sunt apropiate de valoarea label-urilor oferite în setul de date, ele fiind împărțite în diferite subseturi de date.

XGBoost este foarte util în rezolvarea problemelor de clasificare, ceea ce este exact ce proiectul necesită: clasificarea intre două acțiuni pe baza datelor.

În Figura 5 este prezentat modul de funcționare al algoritmului intr-o manieră abstractă.

4 Imprelementarea aspectelor teoretice

În proiect au fost implementate aspectele teoretice relevante, ele având ca scop înțelegerea datelor sau realizarea modelelor în sine.

4.1 Implementarea și Analiza Corelației

Corelația a fost calculată utilizând implementările standard utizilate prezente în librăriile utilizate.

Rezultatul este graficul prezent în Figura 6:

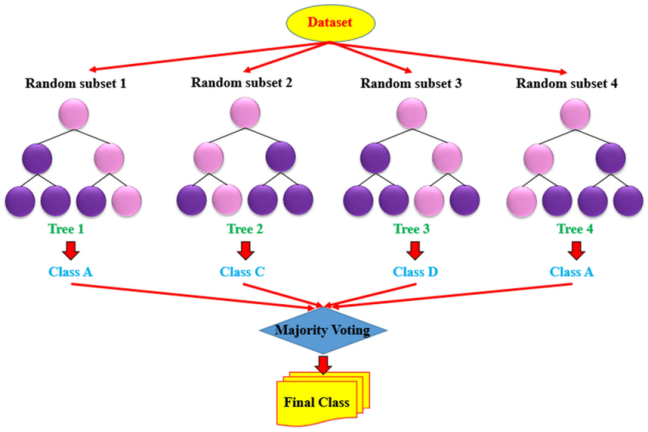

Figure 5: Funcționalitate XGBoost

	wrist	activity	acceleration_x	acceleration_y	acceleration_z	gyro_x	gyro_y	gyro_z
wrist	1.000	-0.113	-0.610	0.087	0.324	-0.022	-0.068	0.009
activity	-0.113	1.000	-0.018	0.640	-0.192	0.041	0.012	-0.008
acceleration_x	-0.610	-0.018	1.000	-0.265	-0.552	-0.022	-0.004	-0.061
acceleration_y	0.087	0.640	-0.265	1.000	0.106	0.011	0.072	-0.023
acceleration_z	0.324	-0.192	-0.552	0.106	1.000	0.035	-0.021	0.050
gyro_x	-0.022	0.041	-0.022	0.011	0.035	1.000	0.094	0.318
gyro_y	-0.068	0.012	-0.004	0.072	-0.021	0.094	1.000	0.287
gyro_z	0.009	-0.008	-0.061	-0.023	0.050	0.318	0.287	1.000

Figure 6: Enter Caption

Din graficul de corelație se pot trage următoarele axe utile pentru înțelegerea datelor:

- Din figura 0 se poate explica de ce există o corelație mare între "wrist" și direcțiile de accelerare pe X și Y.

- "Acceleration_y" și "activity" prezintă o corelație ridicată, implicându-se creșterea accelerației atunci când "activity" este 1 (1 indicând alergat)

- Din rotația telefonului pe încheietură cât și din încheietura pe care a fost ținută se explică și corelațiile între accelerațiile prezente în tabel. ("acceleration_x" cu "acceleartion_z")

- Se observă că datele giroscopice nu au corelația ridicată în legătura cu restul datelor

4.2 Implementare și Analiza Entropiei

În proiect au fost folosite diferite metode de a calcula entropia (în baza e), însă a fost aleasă o metodă generală pentru a calcula entropia.

În setul de date ales, s-a calculat o entropie maximă posibilă de 11.39. Pentru fiecare dintre seturile de coloane cu o importanță mărită rezultând în următoarele rezultate;

- Acceleration_x -¿ 9.93 / 11.39
- Acceleration_y -¿ 9.79 / 11.39
- Acceleration_z -¿ 9.34 / 11.39

Aceste date indică că setul de date are o multitudine de date diferite, ceea ce poate duce spre un model cu o rată de detecție mai ridicată.

4.3 Implementarea Modelelor

Modele au fost implementate utilizând diferite tehnici. Modele de regresie au fost implementate utilizând librăria "sklearn", modelul de LSTM a fost implementat utilizând librăria "tenserflow" iar modelul XGBoost utilizând librăria "xgboost".

Modelele au fost antrenate pe aceleași date, împărțite intre setul de antrenament și setul de validare (împărțirea s-a făcut în proporție de 80/20).

Rezultatele obținute sunt următoarele:

- Regresie liniară : valoarea erori absolute medie : $\tilde{0}.29$

Valorea erori absolute medie este mică, indicând o performanță ridicată.

- Regresie logistică : acuratețea modelului este de $\tilde{8}5\%$

Acuratețea este una ridicată, indicând o performanță ridicată a modelului

-LSTM: valoarea erori absolute medie este de 0.19,

Valoarea erori absolute medie este foarte mică, indicând o performanță foarte ridicată.

-XGBoost: acuratețea modelului este de 99%

Din rezultatele obținute se observă cum modelul clasifică datele cu o acuratețe foarte ridicată, indicând că acest model este aproape perfect în a prezice dacă o persoană se află în fugă sau în mers.

Acest model este cel ales pentru implementare.

5 Testare și validare

5.1 Împărțirea datelor și antrenarea modelului final

Înainte de a se antrena modele este important de menționat că nu toate datele din setul de date inițial nu au fost necesare pentru antrenarea modelului.

Datele ce nu au fost luate în considerare sunt:

-"Username" : datele au fost colectate doar de un utilizator.

- "Time" & "Date" : ora și data pot oferi o direcție greșită modelului, inducând posibilitatea de a prezice o rutină în loc de a prezice o activitate. Articolul realizat de Alexander Martin Mussgnug expune acest predicament în lucrarea sa [9], implicând schimbarea rezultatului final al algoritmului pe baza datelor introduse în antrenarea modelului.

5.2 Antrenarea modelului final

Modelul final a fost antrenat pe setul de antrenament (80% din totalul setului de date) și a fost valid pe setul de validare (20% din totatlul setului de date).

Modelul a fost inițiat folosind "XGBClassifier" inclus în librăria "xgboost", inițiat cu variabilele "random_state" = 42, "verbosity" = 0, "silent" = 0.

După antrenarea modelului a rezultat o acuratețe de 99%.

Modelul a fost reantrenat de mai multe ori pentru a asigura că nu a fost un incident unic. De asemenea se recomandă reantrenarea pe termen lung regulată a modelului pentru a ne asigura că datele coincid cu realitatea, deoarece datele colectate pot fi influențate de factori externi precum vârsta sau sănătatea persoanei responsabile de colectarea datelor

6 Testare manuală și rezultate

6.1 Colectarea datelor pentru testare

Datele de testare au fost colectate în mod asemănător cu metoda de colectare a setului de date inițial. Aplicația utilizată pentru colectare nu a fost facută publică însă folosind aplicația "Sensor Recorder" publicată de Nils Ackermann au fost colectate aceleași date care ar fi fost colectate de aplicația utilizată de autorul setului de date.

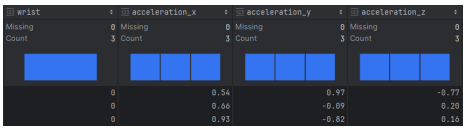

Figure 7: Parte din datele colectate manual

În Figura 7 și Figura 8 se observă o parte din datelele colectate manual ce s-au folosit în testare.

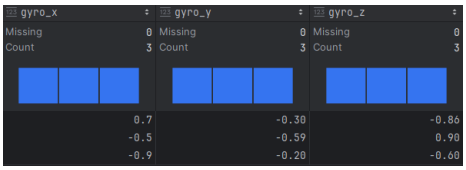

Figure 8: Parte din datele colectate manual

Datele ce au fost colectate sunt alese special pentru a testa dacă sistemul deosbește alergatul (primele două rânduri) de mers (ultimul rând)

Au fost testate aproximativ 50 de instanțe de date în diferite ipostaze, însă aceste 3 au fost alese, ele fiind cele mai apropiate de media datelor.

6.2 Testarea pe date culese manual și validarea modelului

Pe datele culese manual, modelul a prezentat o acuratețe de 100%. Acest lucru se explică prin faptul că setul de date colectate manual este restâns. Chiar dacă setul de date de testare manuală este relativ mic, A Vabalas et al. indică că dacă un set de date este destul de dens, chiar dacă are un număr mic de intrări el poate fi folosit pentru o testare adecvată [10].

În cele 3 instanțe prezente în Figura 7 și Figura 8 rezultatele au fost următoarele : alergat, alergat, mers.

Rezultatele coincid cu activitatea în care ele au fost recoltate.

7 Concluzie

În acest proiect s-a încercat crearea unui model capabil să determine activitatea unei persoane pe baza datelor colectate de senzorul giroscopic cât și a senzorului de accelerație prezenți într-un telefon.

Au fost antrenate diferite modele, fiecare generând un rezultat bun, însă pentru modelul final a fost ales XGBoost, având o rată de acuratețe extrem de ridicată.

Datele colectate au fost curățate în mare parte deja de utilizatorul care le-a înregistrat, însă în acest proiect nu au fost utilizate toate câmpurile, ele având ocazia de a influența negativ modelul final.

În viitor se recomandă implementarea modelului în o aplicație IOS, pentru ca el să poată fi testat în timp real. Acest lucru ar duce la o validare totală a modelului, ce ar putea să influențeze modul în care aplicațiile de urmărie al exercițiilor abordează înregistrarea alergării.

References

[1] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings*

of the 2019 chi conference on human factors in computing systems, pages 1–12, 2019.

[2] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[3] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[4] Ibrahim Ghafir, Mohammad Hammoudeh, Vaclav Prenosil, Liangxiu Han, Robert Hegarty, Khaled Rabie, and Francisco J Aparicio-Navarro. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Generation Computer Systems*, 89:349–359, 2018.

[5] Sunil Kumar and Ilyoung Chong. Correlation analysis to identify the effective data in machine learning: Prediction of depressive disorder and emotion states. *International journal of environmental research and public health*, 15(12):2907, 2018.

[6] Gülden Kaya Uyanık and Neşe Güler. A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences*, 106:234–240, 2013.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Didrik Nielsen. Tree boosting with xgboost-why does xgboost win" every" machine learning competition? Master's thesis, NTNU, 2016.

[9] Alexander Martin Mussgnug. The predictive reframing of machine learning applications: good predictions and bad measurements. *European Journal for Philosophy of Science*, 12(3):55, 2022.

[10] Andrius Vabalas, Emma Gowen, Ellen Poliakoff, and Alexander J Casson. Machine learning algorithm validation with a limited sample size. *PloS one*, 14(11):e0224365, 2019.