Water quality analysis using machine learning

COSTIN RADU IONUȚ
UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA
Facultatea de Inginerie
Specializarea: Calculatoare
Email: Costin.Ra.Radu@student.utcluj.ro

Abstract
This paper discusses a method for developing a process to analyze the water quality based on recorded data. This is an attempt to reduce water pollution and warn people in areas with non-potable water to avoid consuming it. The project is carried out in RapidMiner Studio application for finding out which attribute influences the quality of water. Further, various machine learning algorithms were trained to achieve the best possible results, and the most suitable algorithms were selected based on their performance metrics and to be used in an ensemble voting algorithm.
Keywords: *RapidMiner, Machine Learning, Naive Bayes, KNN, Random Forest, Classification, SVM.*

Introducere

Accesul la apă potabilă sigură este esențial pentru sănătate, un drept fundamental al omului și o componentă a politicii eficiente de protecție a sănătății [1], așadar este necesar să știm cât de sigură este apa care ne intră în organism. Acest proiect folosește tehnici de învățare automată, implementate în RapidMiner, pentru a afla cât de sigură este apa și care este cel mai important factor care influențează potabilitatea.

Principalul scop al proiectului este aflarea factorilor care influențează rezultatul final de apreciere a potabilității surselor de apă și implementarea unor algoritmi de învățare automată pentru această problemă de clasificare, urmărind pricipiile de comparare a performanțelor obținute. Ulterior, vor fi aleși și folosiți trei dintre ei în crearea unui sistem de voting, pentru validatea modelului implementat.

Problemele de clasificare reprezintă unul din capitolele majore al domeniului data mining [2]. Practic, conform Umadevi și Marseline [3], modelele de clasificare sunt antrenate pentru a prezice un anumit rezultat bazat pe niște date de intrare. Problemele de clasificare sunt deseori folosite pentru a rezolva probleme precum:

- Predictia comportamentului clientului [4];
- Filtru de spam [5];
- Clasificare de imagini [6];
- Clasificarea textului dintr-o pagină web [7];
- Categorizarea produselor;
- Clasificarea Malwarului [8];
- Predicția abandonului clienților;
- Detecția fraudei cu card de credit [9];
- Analiza sentimentelor. [10]

Clasificarea calității apei folosind algoritmi de învățare automată a fost abordată de diferiți cercetători, din perspective diferite, printre care amintim:

- Nasir et al. [11], clasifică apa folosind algoritmi de învățare automată;
- Rana et al. [12] analizează influențele calității apei asupra mediului înconjurător.

ISSN

Implementare şi rezultate ponderi

Setul de date preluat de pe Kaggle[1] conţine valori pentru calitatea apei pentru 3276 de corpuri de apă diferite şi decide dacă apa este potabilă sau nu în funcţie de: valoarea pH-ului; duritatea; totalul solidelor dizolvate (TDS); cloraminele; sulfat; conductivitate; carobonul organic total (TOC); trihalomethane şi turbiditate. După introducerea datelor în RapidMiner, s-a construit un proces (Fig. 1.) prin care s-a aflat corelaţia, câştigul de informaţie, rata câştigului de informaţie şi indexul gini, apoi s-au agregat şi am putut afla din rezultatele obţinute (Fig. 2.) că sulfatul afectează cel mai mult potabiliatea apei.

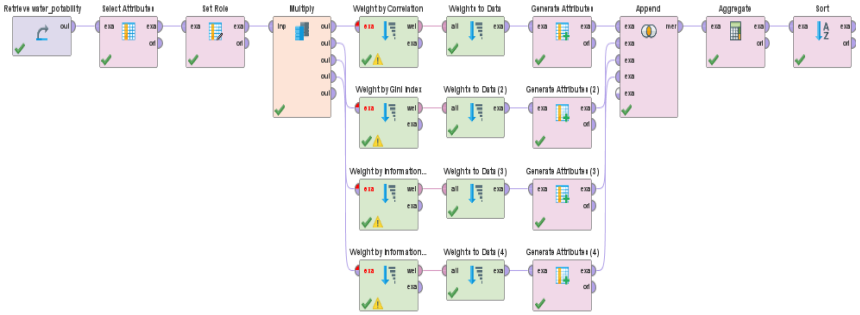

Fig. 1. Implementarea ponderilor

Row No.	Attribute	sum(Weight)
1	Sulfate	0.169
2	Hardness	0.157
3	Chloramines	0.146
4	Organic_carb...	0.137
5	Solids	0.096
6	Conductivity	0.069
7	ph	0.064
8	Trihalometha...	0.062
9	Turbidity	0.061

Fig. 2. Rezultate ponderi

Implementare şi rezultate procese

În următoarea etapă, am ales şase algoritmi de clasificare care pot funcţiona cu tip de date numeric şi cu date lipsă, adică:

- Decision Tree, care împarte datele în mod recursiv în vederea obţinerii atributelor până în punctul în care apare o condiţie de oprire. Ramura Arborelui Decizional reprezintă condiţia testului, nodurile de decizie descriu proprietăţile, iar nodurile frunzelor reprezintă etichetele clasei [13];
- Random Forest este un algoritm de învăţare automată care este utilizat la scară largă, el construieşte arbori de decizie pe diferite eşantioane şi le ia votul majoritar pentru clasificare. Potrivit lui Breiman [14], cea mai importantă caracteristică a algoritmului este că poate gestiona seturi de date care conţin variabile categoriale;
- K-nn, despre care Robnik-Šikonja şi Igor [15] au spus că este un algoritm leneş de învăţare supervizată utilizat în clasificare şi regresie care încearcă să clasifice în clasa corectă prin calcularea distanţei euclidiene dintre datele de testare şi toate punctele de antrenare;
- Logistic Regression este un algoritm de clasificare utilizat pentru a prezice probabilitatea unei variabile ţintă. Natura variabilei ţintă sau dependentă este dihotomică [16];
- Support Vector Machine, conform Ukil [17] are scopul de a crea cea mai bună linie sau graniţă de decizie care poate segrega spaţiul n-dimensional în clase, astfel încât să putem pune cu uşurinţă noul punct de date în categoria corectă în viitor;
- Naive Bayes, care reprezintă „clasificatori probabilistici” simpli, bazaţi în primul rând pe utilizarea teoremei lui Bayes, cu ipoteze puternice "naive" de independenţă, aşa cum a afirmat Zhang [18].

[1] https://www.kaggle.com/datasets/adityakadiwal/water-potability

În etapa de preprocesare (Fig. 3) înainte de implementarea algoritmilor din procesul de optimizare şi selecţie, după ce setul de date a fost extras, s-a folosit parametrul “Remove duplicates”, cu scopul de a înlătura valorile identice; “Set role”, pentru a schimba rolul atributului “Potability” în label (atribut de prezis) şi am aplicat un filtru asupra datelor utilizând “Filter Examples”, care înlătură coloanele unde atributul label lipseşte. Setul de date a fost pe urmă multiplicat pentru a fi trimis componentelor următoare.

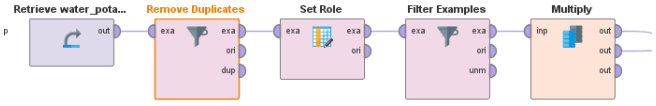

Fig. 3. Preprocesare date

Datele au fost împărţite astfel: 70% pentru antrenare şi 30% folosind „Cross Validation” (Fig. 4.), care permite execuţia în paralel. Algoritmii au fost implementaţi în partea de training, iar modelul aplicat asupra a 30% din setul de date din partea de testing(Fig. 5.), iar cele mai bune perfomanţe înregistrate sunt prezentate în Fig. 11-13.

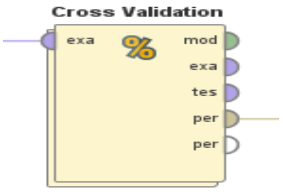

Fig. 4. Cross Validation

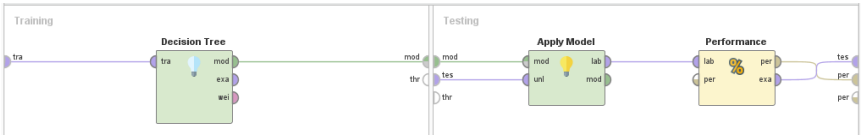

Fig. 5. Exemplu Training şi Testing în cross validare

Fiecare “Cross Validation” a fost pus într-un “Optimize parameters”(Fig. 6.), care execută toate combinaţiile de valori ale parametrilor selectaţi (fiecare algoritm a avut alţi parametrii) şi apoi furnizează rezultatele fiecărei iteraţii, iar în acelaşi timp, am legat operatorul “Remember” la cel de optimizare, pentru a stoca datele rezultate. Implementarea algoritmilor din interiorul parametrului de optimizare, cât şi “Remember” (Fig. 7.) a fost făcută apoi într-un subproces (Fig. 8.) din interiorul unui alt “Optimize parameters” pentru rularea a şase fire de execuţie, cu fiecare algoritm în acelaşi timp, iar ca să ne asigurăm că nu luăm numai performanţele, cât şi ce model a fost selectat, s-a conectat şi outputul „par” la rezultat, aşadar s-a văzut cea mai bună acurateţe şi timp de execuţie.

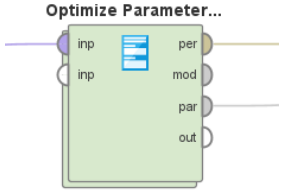

Fig. 6. Optimize Parameters

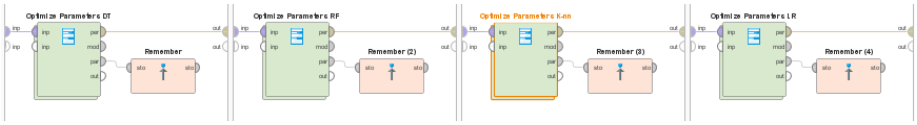

Fig. 7. Plasare operatori în Subproces

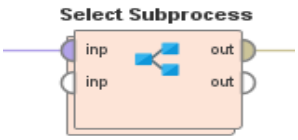

Fig. 8. Subprproces

Ulterior, setul de date a fost multiplicat, iar rezultatele memorate de către "Remember" au fost trimise împreună cu setul de date către "Compare ROCs" (Fig. 9.), unde cu ajutorul lui Recall şi Set parameters, alături de algoritmul aferent au fost desenate curbe ROC pentru fiecare algoritm, folosind datele din optimizare (Fig. 10.), astfel putându-se observa grafic performanţele procesului de optimizare şi selecţie.

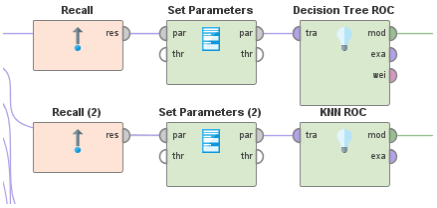

Fig. 9. Exemplu interior Compare ROC's

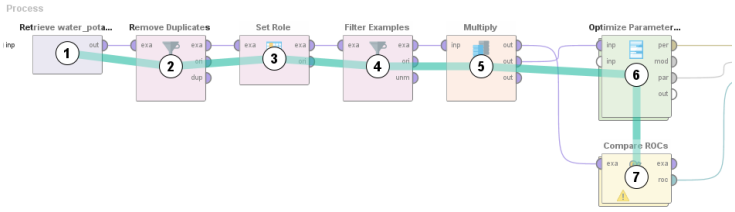

Fig. 10. Trimitere date către Optimize Parameters şi Compare Roc's

După ce s-a observat cea mai bună acurateţe din fiecare algoritm, ţinând cont de acurateţe, timp, compatibilitate cu datele şi cel mai important, reprezentarea vizuală (Fig. 14.), s-au ales trei algoritmi, cu cele mai bune setări, pentru a fi aplicaţi într-un algoritm de votare.

Algoritmii aleşi sunt Naive Bayes: acurateţe 0,623 (Fig. 12.), Random Forest: acurateţe 0,615 (Fig. 13.), respectiv Logistic Regression: acurateţe 0,612 (Fig. 11.).

iteration	Cross V...	accuracy
1	2	0.612
2	12	0.610
3	22	0.610
6	51	0.610
4	31	0.611
7	61	0.610
8	71	0.610
5	41	0.610
10	90	0.610
9	80	0.610
11	100	0.610

Fig. 11. Acurateţe Logistic Regression

iteration	Naive B...	accuracy
2	false	0.623
1	true	0.622

Fig. 12. Acurateţe Naive Bayes

iteration	Rando...	accuracy	precision	recall	AUC (op...	AUC	AUC (pe...
1	1	0.614	0.750	0.014	0.997	0.500	0.014
2	2	0.614	0.800	0.016	0.998	0.506	0.016
3	3	0.615	0.667	0.025	0.992	0.509	0.025
4	4	0.613	0.658	0.019	0.993	0.506	0.019
7	7	0.614	0.706	0.019	0.989	0.507	0.025
6	6	0.614	0.857	0.017	0.990	0.509	0.027
8	8	0.614	0.760	0.015	0.990	0.510	0.029
5	5	0.614	0.700	0.027	0.990	0.509	0.027
9	9	0.612	0.700	0.017	0.990	0.509	0.028
10	10	0.616	0.744	0.023	0.990	0.510	0.031

Fig. 13. Acurateţe Random Forest

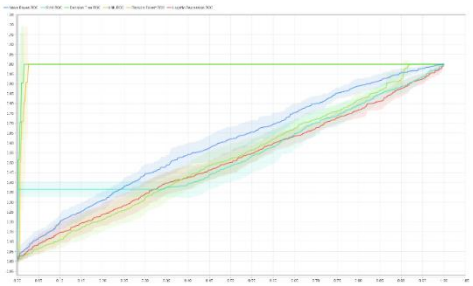

Fig. 14. Grafic curbe ROC

Așa cum spunea Tsymbal [19], ensemble learning este o metodă prin care se caută cea mai bună precizie clasificatoare prin alăturarea mai multor modele de învățare automată. Una dintre aceste metode ensamble este algoritmul de votare.

Desprins din lucrarea lui Ruta și a lui Gabrys [20], algoritmul de votare antrenează mai multe modele, unde majoritatea se supune minorității, cu scopul de a ajunge la un rezultat mai bun.

Pentru început, setul de date a fost extras și rolul atributului "Potability" setat ca label folosind "Set Role", setul de date a fost filtrat, în așa fel încât să nu existe label lipsă. Folosind un Cross Validation, setul de date a fost împărțit pentru training și testing (Fig. 15.). În partea de training, a fost aplicat algoritmul de voting, în care au fost puși algoritmii aleși, iar în partea de testare, s-a aplicat modelul antrenat (Fig. 16.) folosind cei 3 algoritmi (Fig. 17.) și performanța calculată ca fiind 61,42% (Fig. 18.).

Deoarece se dorește aplicarea modelului pe un set de date nou, orice set nou de date poate fi introdus în aplicație și conectat la inputul „uni", apoi se aplică modelul antrenat, iar o parte din rezultatele obținute sunt prezentate în Fig. 19.

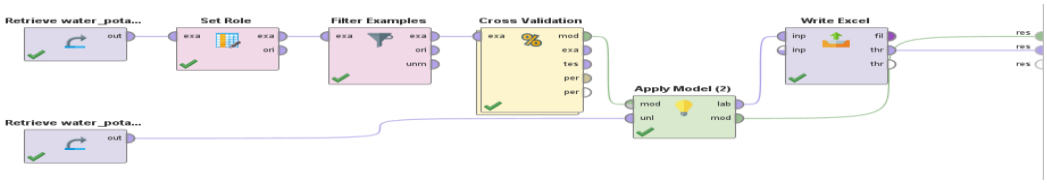

Fig. 15. Verificarea potabilității apei

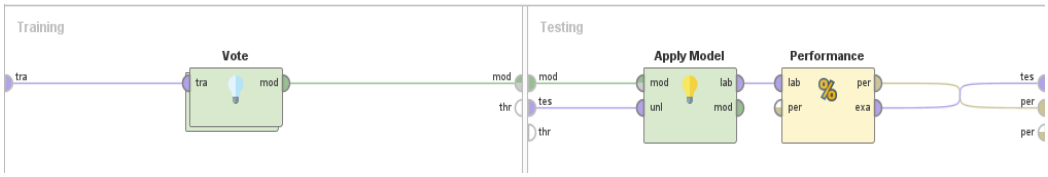

Fig. 16. Implemenare algoritm voting

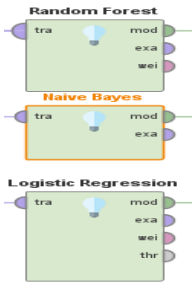

Fig. 17. Algoritmii din voting

accuracy: 61.42% +/- 0.57% (micro average: 61.42%)

	true no	true yes	class precision
pred. no	1977	1243	61.40%
pred. yes	21	35	62.50%
class recall	98.95%	2.74%	

Fig. 18. Acurateţe algoritm de votare

24	no	1	0	7.774	216.753	22316.398	7.948	385.043	288.069	14.137	68.862	3.891
25	yes	0.333	0.667	?	172.731	28894.477	4.251	244.285	559.629	21.341	51.275	2.960
26	no	1	0	7.203	168.445	22826.485	6.283	271.892	437.371	16.411	64.506	6.389
27	no	0.667	0.333	6.150	240.888	6342.503	9.610	421.343	354.393	12.340	83.239	3.552
28	no	0.667	0.333	?	225.959	9660.658	8.828	397.924	315.751	16.717	17.001	3.959
29	yes	0.333	0.667	6.768	203.751	21162.727	5.233	251.062	334.905	13.214	79.863	3.300
30	no	0.667	0.333	8.037	148.415	48410.471	4.756	268.212	392.901	12.467	?	2.506
31	no	1	0	5.344	198.379	13492.841	6.559	328.649	591.363	14.084	61.393	4.105

Fig. 19. Rezultate ale procesului de voting pe un set nou de date

Concluzii

În urma derulării acestui proiect am putut observa cât de impresionantă este capacitatea de a prognoza potabilitatea apei şi am identificat ponderea în care factorii influenţează rezultatul, aflând că sulfatul afectează cel mai mult. De asemenea, am remarcat în urma unui proces complex de optimizare că cei mai buni algoritmi de învăţare automată care pot clasifica potabilitatea apei, aplicaţi pe acest set de date sunt Naive Bayes, Random Forest şi Logistic Regression, care odată implementaţi într-un proces de votare reuşesc să antreneze un model eficient pentru a garanta reuşita problemei de clasificare.

Bibliografie

[1] G. Ion, „Dreptul la apă–un nou drept fundamental al omului.,” *Revista de Ştiinţă, Inovare, Cultură şi Artă „Akademos,* vol. 18, nr. 3, pp. 39-45, 6 Sept 2010.

[2] N. S. S, „A comparative study of classification techniques in data mining algorithms,” *Oriental Journal of Computer Science and Technology,* vol. 8, nr. 1, pp. 13-19, 30 Apr 2015.

[3] S. a. K. J. M. Umadevi, „A survey on data mining classification algorithms.,” *International Conference on Signal Processing and Communication,* pp. 264-268, 28 Jul 2017.

[4] J. S. P. a. L. H. Li, A machine learning based method for customer behavior prediction, 2019, pp. 1670-1676.

[5] S. K. a. N. B. Tuteja, „Email Spam filtering using BPNN classification algorithm.,” în *nternational Conference on Automatic Control and Dynamic Optimization Techniques*, 2016.

[6] D. a. Q. W. Lu, „A survey of image classification methods and techniques for improving classification performance,” *International journal of Remote sensing,,* vol. 28, nr. 5, pp. 823-870, 1 Mar 2007.

[7] Z. a. Z. Q. Wang, „Research on Web text classification algorithm based on improved CNN and SVM.,” în *17th International Conference on Communication Technology*, 2017.

[8] G. E. S. J. W. D. L. &. Y. D. Dahl, „Large-scale malware classification using random projections and neural networks.," în *International Conference on Acoustics, Speech and Signal Processing*., 2013.

[9] A. R. T. a. Y. D. Shen, „Application of classification models on credit card fraud detection.," în *2007 International conference on service systems and service management*, 2007.

[10] A. S. J. P. G. A. K. a. A. G. Gupte, „Comparative study of classification algorithms used in sentiment analysis.," *International Journal of Computer Science and Information Technologies,* vol. 5, nr. 5, pp. 6261-6264, 2014.

[11] N. A. K. O. A. F. B. M. S. A. S. a. A. A.-S. Nasir, „Water quality classification using machine learning algorithms," *Journal of Water Process Engineering,* 22 Aug 2022.

[12] M. a. R. A. a. D. J. a. A. S. a. M. J. a. P. B. Rana, „Machine learning approach to investigate the influence of water quality on aquatic livestock in freshwater ponds," *Biosystems Engineering2021,* vol. 208, pp. 164-175, 2021.

[13] E. ş. O. E. Paraschiv, „Tehnici bazate pe Machine Learning pentru îmbunătăţirea depistării cancerului de sân," *Romanian Journal of Information Technology and Automatic Control,* vol. 30, nr. 2, pp. 67-80, 2020.

[14] L. Breiman, „Random forests," *Machine learning,* vol. 45, nr. 1, pp. 5-32, 45 Oct 2001.

[15] M. a. I. K. Robnik-Šikonja, „Explaining classifications for individual instances," *Transactions on Knowledge and Data Engineering,* vol. 20, nr. 5, pp. 589-600, 31 Mar 2008.

[16] J. S. Cramer, „The origins of logistic regression.," 2002.

[17] A. Ukil, Intelligent Systems and Signal Processing in Power Engineering, Springer, 2007, pp. 161-226.

[18] Z. Harry, „The optimality of naive Bayes.," *Aa,* vol. 1, nr. 2, 2004.

[19] A. Tsymbal, „The problem of concept drift: definitions and related work," *Computer Science Department, Trinity College Dublin,* vol. 106, nr. 2, p. 58, 2004.

[20] D. a. G. B. Ruta, „Classifier selection for majority voting," *Information fusion,* vol. 6, nr. 1, pp. 63-81, 2005.