Proiect Funadamente de Big-data

Calitatea apei proaspete

Dragotă Iulia-Mirela

Seician Aurel

Informatică Economica anul 3

Cuprins:

Introducere

Calitatea apei este esentiala in prezent , iar in urma unei analize amanuntite aceasta ne asigura anumite masuri de siguranta a consumului. In studiul nostru dorim sa aflam potabilitatea apei, ceea ce inseamna ca apa cu cea mai buna calitate poate sa fie consumata fara a prezenta reactii adverse. Din pacate exista destul de multe zone in care oamenii au un access restrans la apa potabila, iar acest lucru ii face sa recurga la uzul apei contaminate cu diferite substante care pe termen lung au efecte negative asupra corpului, atat pe interior, cat si pe exterior, iar cel mai grav este faptul ca o sursa de apa contaminate are puterea e a raspandi diverse boli. UNICEF a ajuns la concluzia ca, la nivel Mondial, unu din cinci copii nu are suficienta apa pentru acoperirea nevoilor sale zilnice. Conform datelor disponibile, există peste 80 de țări în care copiii se confruntă cu probleme ridicate sau extrem de ridicate legate de accesul la apă. Cele mai afectate zone sunt Africa de Est si de Sud, Africa de Vest si Centrala, Asia de Sud si Orientul Mijlociu. Numarul total de copii care traiesc in zone cu o vulnerabilitate hidrica ridicata sau extrem de ridicata in Asia de Sud este de peste 155 de milioane.

Calitatea apei in zonele rurale si urbane reprezinta o problema serioasa, iar rezolvarea acesteia aduce multe beneficii regiunilor si are un impact economic pozitiv, deoarece studiile arata ca o calitate scazuta a vietii produce costuri mai ridicate in domeniul sanatatii, iar apa contaminata raspandeste bolile.

Asadar, problema principala a analizei noastre se refera la evaluarea calitatii apei proaspete. In acest proiect dorim sa raspundem la intrebarea urmatoare: **In ce masura putem sa prezicem potabilitatea apei in functie de substantele existente in ea?**

Setul de date

Structura:

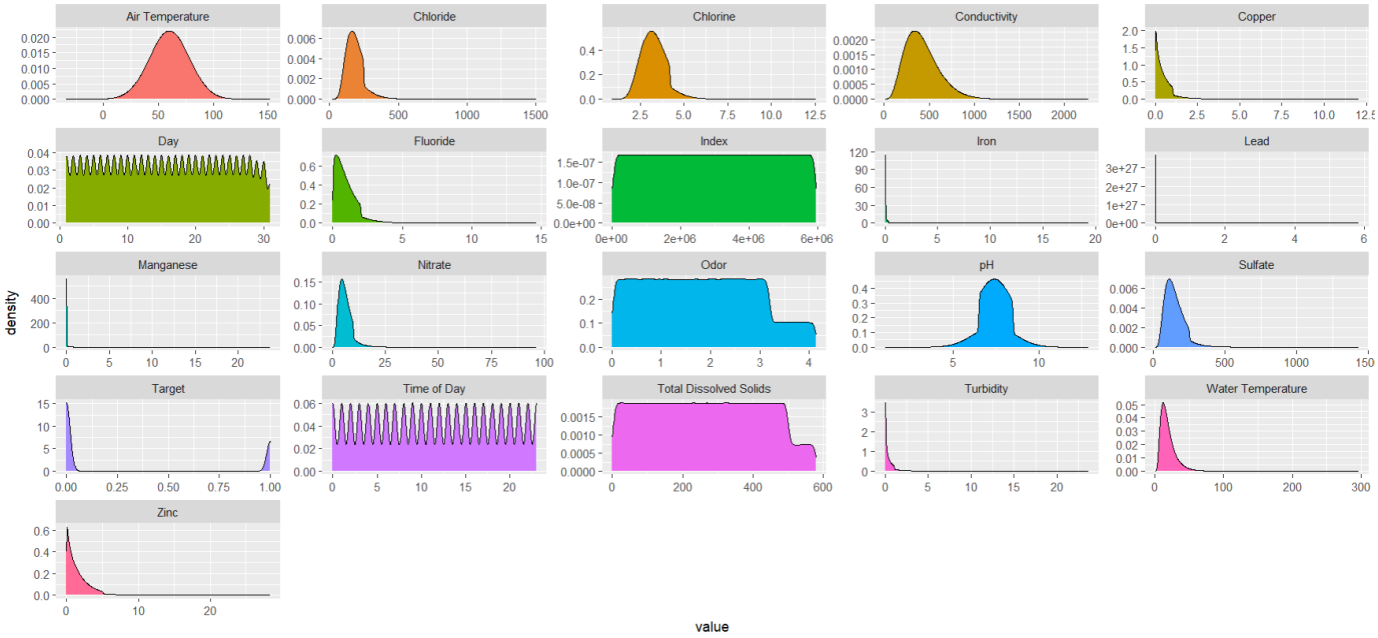
Pentru realizarea proiectului am folosit baza de date gasita pe linkul https://www.kaggle.com/datasets/naiborhujosua/predict-the-quality-of-freshwater . Baza de date prezinta peste 5 milioane de inregistrari fiind formata din 24 coloane care reprezinta factorii care pot influenta calitatea apei si o coloana „Target” care indica daca apa este sau nu potabila cu ajutorul cifrelor 0 care reprezinta nepotabila si 1 care reprezinta potabila.

Am curatat si modificat datele, ceea ce inseamna ca am eliminat inregistrarile lipsa(NA), ajungand la un set de 3981800, iar apoi am decis sa lucram cu un set mai mic de date , mai exact 10000 de inregistrari. La un moment dat am factorizat atributul ‚Target’ in felul urmator: valorile 0 (nepotabila) si 1(potabila) au fost inlocuite cu ‚No’ si ‚Yes’, astfel avand urmatorul numar de inregistrari:

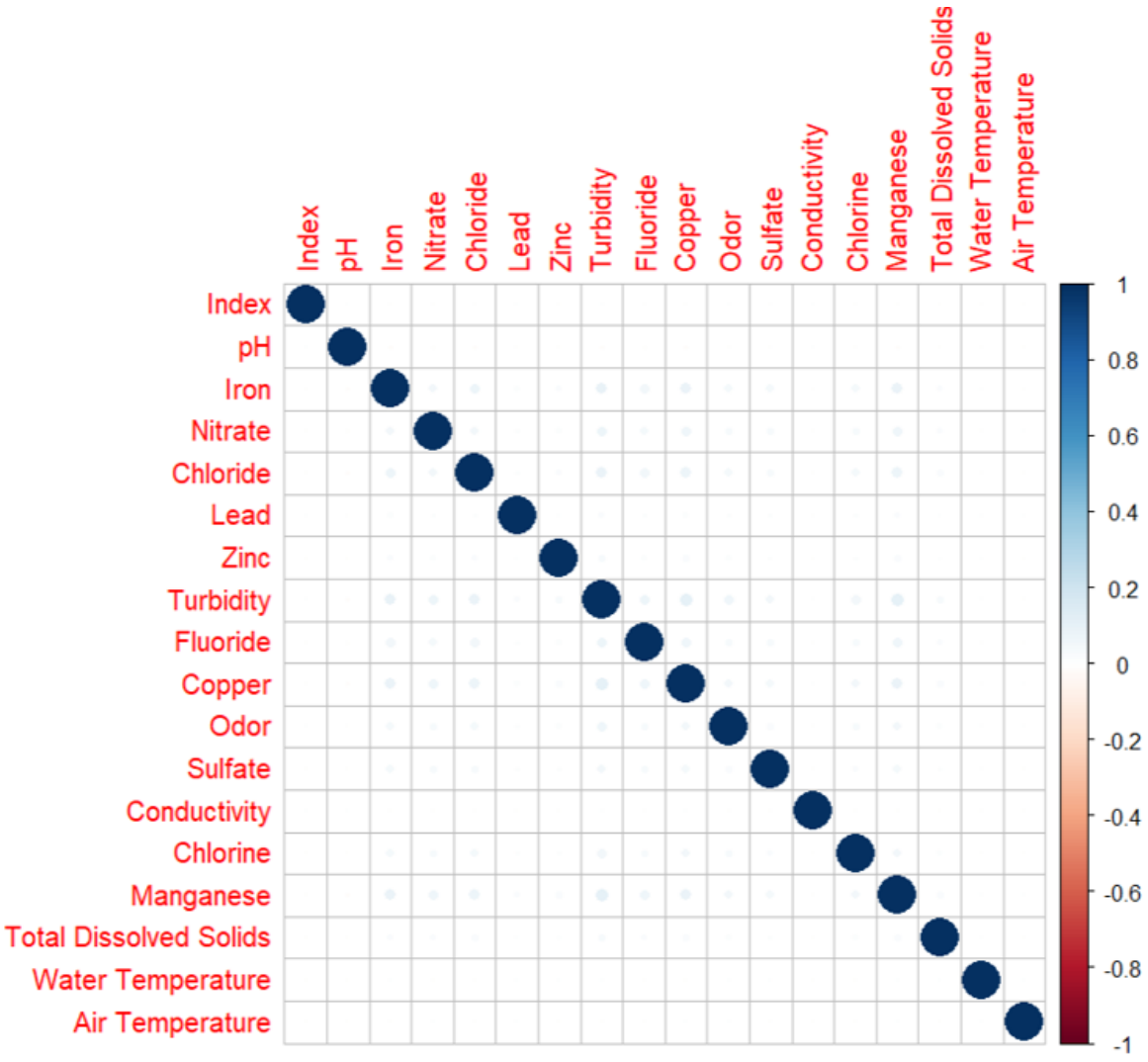
0	1
6926	3074

2. Vizualizare

Mai jos este un set de grafice de densitate pe care l-am creat pentru fiecare variabila numerica din setul nostru de date si arata distributia valorilor pentru fiecare variabila numerica in mod individual.



Pentru realizarea modelului am creat si o matrice de corelatie a variabilelor independente:



Analiza Datelor

Intrebarea care ne intereseaza, ci anume : Este apa potabila in functie de substantele din ea? Aceasta reprezinta o problema de clasificare, iar metodele principale la care am apelat din Machine Learning sunt: Regresia logistica ,Naïve Bayes si Arbori de decizie. Fiecare metoda a fost optimizata, iar apoi acestea au fost comparate intre ele pentru a putea alege cel mai bun model.

Dup ace am curatat setul de date, l-am impratit intr-un set de antrenament si unul de test, pe baza proportiei de 70%-30%, iar distributia valorilor a fost mentinuta cu strata = 'Target'. Variabilele independente au rol de predictor si sunt notate cu 'x', iar valorile variabilei dependente cu 'y' si vor fi folosite la antrenarea 'caret'.

1. **Regresie logistica**

Prima metodă de analiză aleasă este regresia logistică. Am început prin a realiza câte un model individual pentru fiecare atribut, pentru a observa relevanța acestuia în raport cu variabila dependentă.

Am construit primul model utilizand toate variabilele din setul de date. Am utilizat algoritmul 'glm' pentru a putea face predictii pe variabila de iesire, folosind distributia binomiala, deoarece variabila de iesire este binara. De asemenea am aplicat cross-validation cu numarul de fold-uri setat la 5. Dup ace am efectuat predictii pe setul de testse calculeaza matricea de confuzie.

```
fitControl <- trainControl(method = 'cv', number = 5)
model <- train(
  x=x,
  y=y,
  method="glm",
  family="binomial",
  trControl = fitControl
)
predictions <- predict(model, newdata = test)
conf1<-confusionMatrix(factor(ifelse(predictions > 0.5, 1, 0), levels = c(0,1)), factor(test$Target, levels = c(0, 1)))
```

Al doilea model utilizeaza doar un subset de variabile, ci anume: Clorhide, Tubidity si Copper. Am utilizat in continuare algoritmul 'glm' cu distributie binomiala. In continuare am facut predictii si am calculate matricea de confuzie.

```
model2 <- glm(Target ~ Chloride + Turbidity + Copper, family = binomial, data = train)
pred2 <- predict(model2, newdata = test, type = "response")
conf2 <-confusionMatrix(factor(ifelse(pred2 > 0.5, 1, 0), levels = c(0,1)), factor(test$Target, levels = c(0, 1)))
```

Cel de al treilea model si ultimul utilizeaza o singura variabila: Copper si se utilizeaza algoritmul ‘glm’ cu distributia binomiala.

```
model3 <- glm(Target ~ Copper, family = binomial, data = train)
pred3 <- predict(model3, newdata = test, type = "response")
conf3 <-confusionMatrix(factor(ifelse(pred3 > 0.5, 1, 0), levels = c(0,1)), factor(test$Target, levels = c(0, 1)))
```

In continuare am calculate metrici de performanta pentru fiecare model in parte, inclusive acuratetea, sensibilitatea si specificitatea. Acestia sunt mai departe afisati in consola.

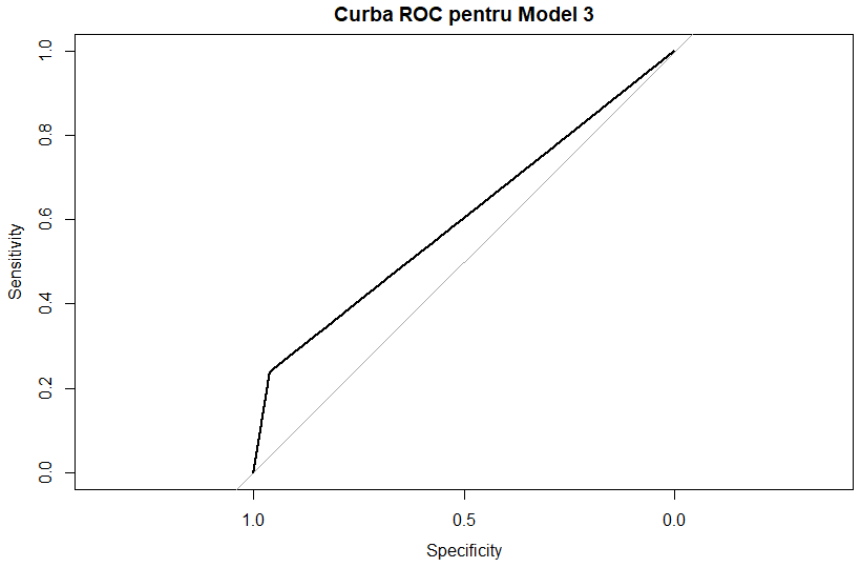
```
metrics <- data.frame(
  model = c("Model 1", "Model 2", "Model 3"),
  accuracy = c(conf1$overall["Accuracy"], conf2$overall["Accuracy"], conf3$overall["Accuracy"]),
  sensitivity = c(conf1$byClass["Sensitivity"], conf2$byClass["Sensitivity"], conf3$byClass["Sensitivity"]),
  specificity = c(conf1$byClass["Specificity"], conf2$byClass["Specificity"], conf3$byClass["Specificity"])
)

# Format metrics
metrics$accuracy <- sprintf("%.2f%%", metrics$accuracy * 100)
metrics$sensitivity <- sprintf("%.2f%%", metrics$sensitivity * 100)
metrics$specificity <- sprintf("%.2f%%", metrics$specificity * 100)
```

Cu ajutorul tabelului comparative putem vizualiza diferentele dintre cele 3 matrici obtinute:

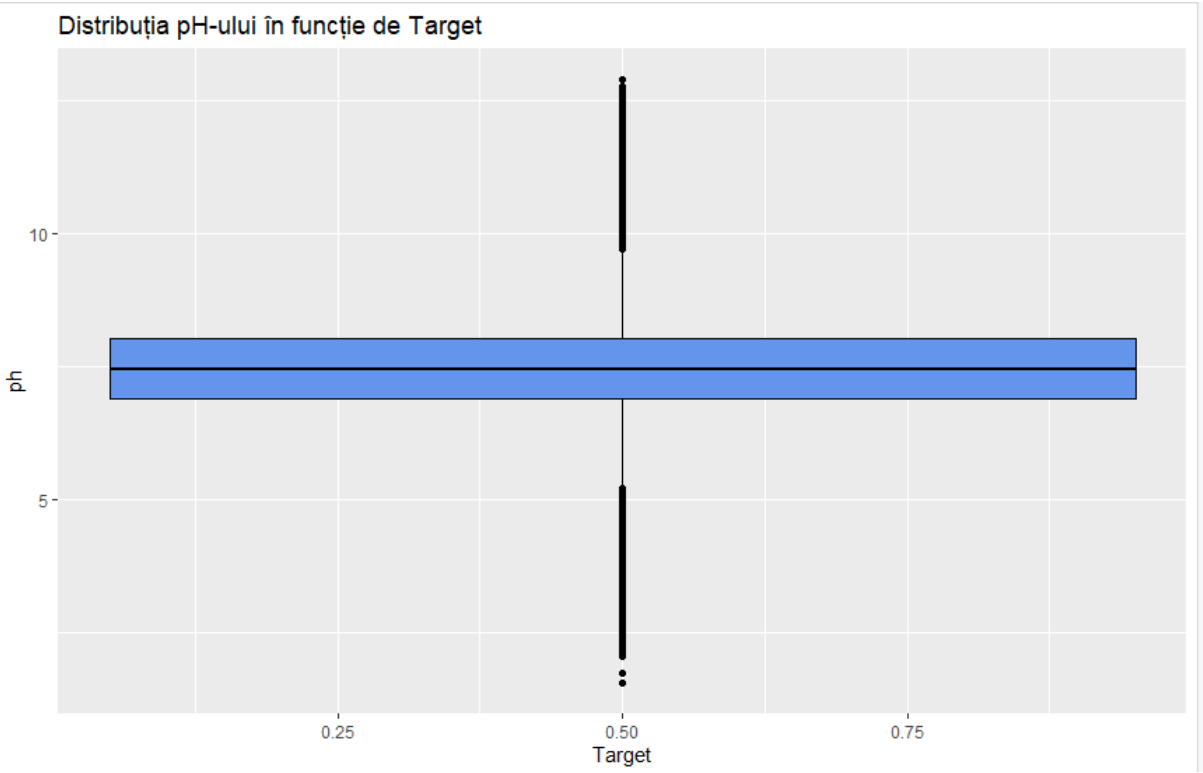
	model	accuracy	sensitivity	specificity
1	Model 1	79.84%	91.96%	51.50%
2	Model 2	75.51%	95.86%	27.92%
3	Model 3	72.68%	97.95%	13.57%

Concluzia pe care o putem trage din acest tabel este că Modelul 1 are cea mai bună performanță generală (cu o acuratețe de 79.84%), dar are o specificitate mai scăzută decât Modelul 2 și Modelul 3. Modelul 2 are cea mai mare sensibilitate (95.86%), dar o specificitate foarte scăzută (27.92%). În schimb, Modelul 3 are o sensibilitate foarte ridicată (97.95%), dar o specificitate extrem de scăzută (13.57%). Prin urmare, concluzia ar fi că Modelul 1 este cel mai echilibrat dintre cele trei modele, cu o performanță generală bună și o sensibilitate și specificitate decente.

Am create si curba ROC care ilustrează performanța modelului de clasificare în funcție de pragul de decizie. Aceasta reprezintă o comparație între rata de detectare a claselor pozitive și rata de detectare a claselor negative la diferite praguri de decizie.




```
Area under the curve: 0.5998
```



2. **Naïve bayes**

Acest model se bazează pe teoria probabilităților condiționate și presupune independența între variabile. În exemplul nostru în care se analizează calitatea apei, modelul va examina probabilitatea ca apa să fie potabilă sau nepotabilă în funcție de valorile fiecărui element chimic.

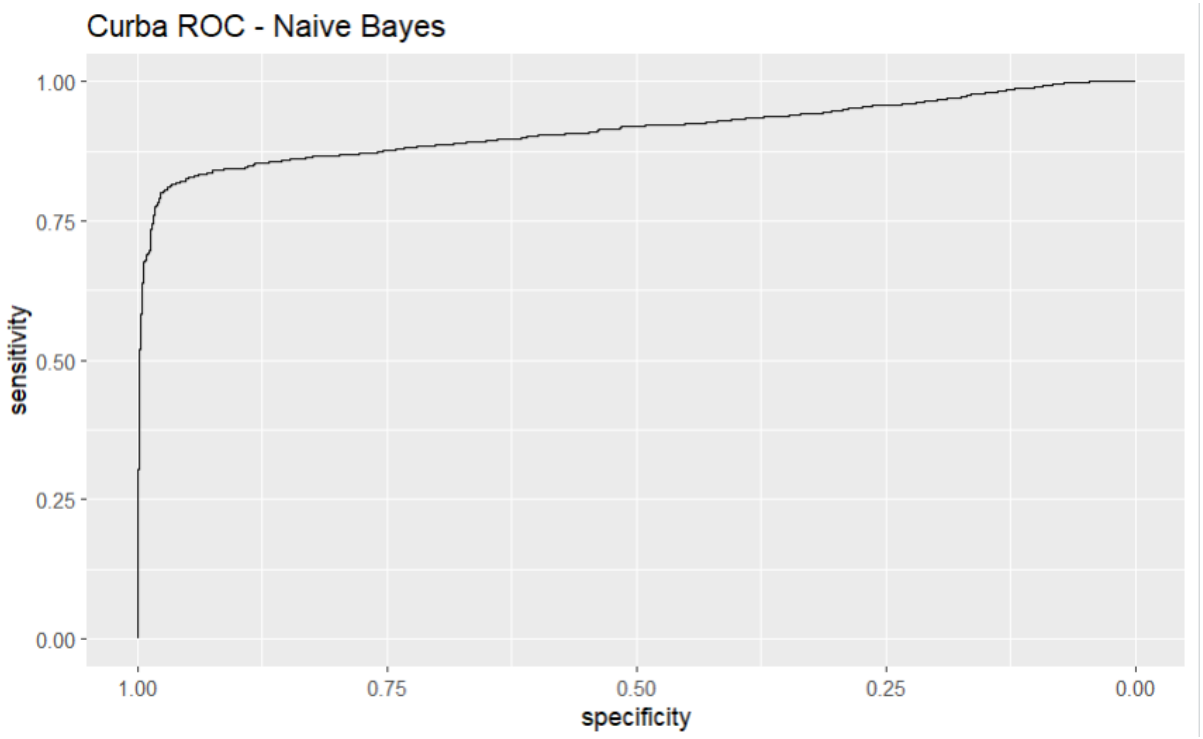
usekernel	ROC	Sens	Spec
FALSE	0.8958599	0.5178705	0.9169807
TRUE	0.9083514	0.2523450	0.9597417

Valorile afișate reprezintă rezultatele metricilor de evaluare pentru modelul antrenat cu două setări diferite pentru hiperparametrul "usekernel": FALSE și TRUE.

ROC este o măsură a performanței generale a modelului, care arată capacitatea acestuia de a face distincție între clase. Cu cât valoarea ROC este mai mare, cu atât modelul are o performanță mai bună.

Sens reprezintă capacitatea modelului de a detecta corect clasele pozitive. Cu cât valoarea sensibilității este mai mare, cu atât modelul este mai bun la detectarea corectă a claselor positive

Spec reprezintă capacitatea modelului de a detecta corect clasele negative. Cu cât valoarea specificității este mai mare, cu atât modelul este mai bun la detectarea corectă a claselor negative.



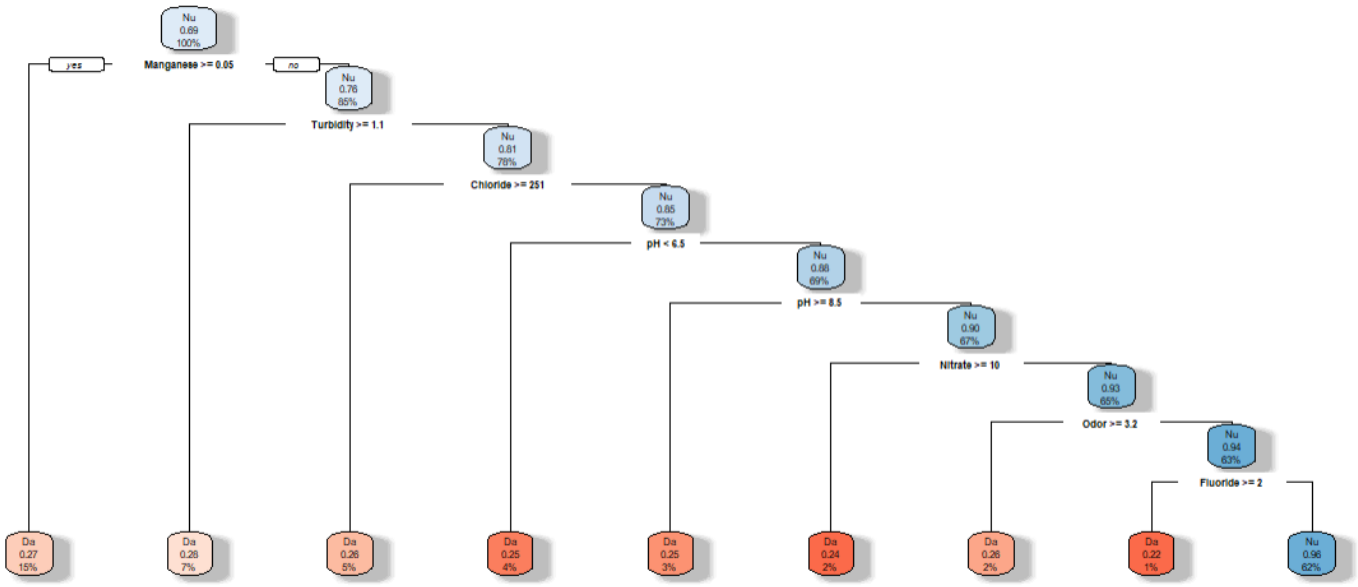
Graficul arată cât de bine se descurcă modelul în separarea claselor pozitive de cele negative. Cu cât curba ROC este mai aproape de colțul din stânga-sus al graficului, cu atât modelul are o performanță mai bună.

```
Area under the curve: 0.9134
```

Aria de sub curba reprezintă o măsură a performanței globale a modelului. O valoare mai mare indică o performanță mai bună a modelului. Aceasta variază între 0 și 1, unde o valoare de 0.5 indică o performanță aleatorie, iar o valoare de 1 indică o performanță perfectă de clasificare.

3. **Arbori de decizie**

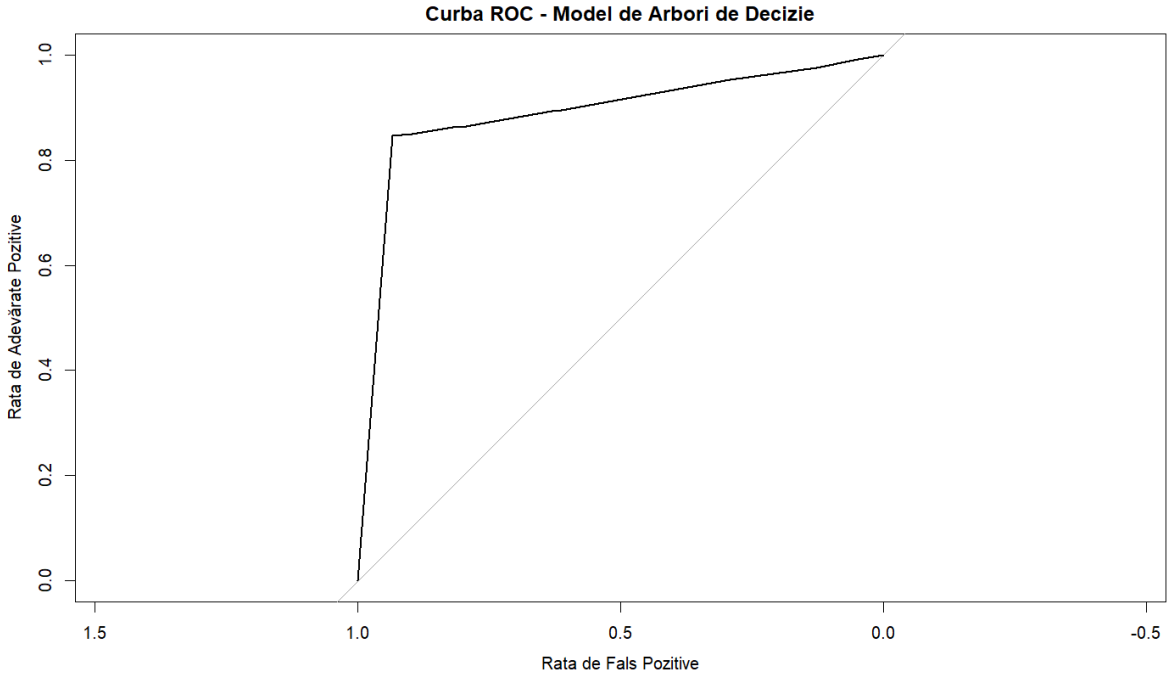
Vom utiliza funcția rpart pentru a construi un model de arbori de decizie. Acest model va fi antrenat pe setul de antrenament numit "train", unde variabila țintă este denumită "Target", iar toate celelalte variabile disponibile vor fi luate în considerare pentru procesul de antrenare.



Calculăm acuratețea modelului prin compararea predicțiilor cu valorile reale din setul de testare. Acuratețea este definită ca numărul de predicții corecte împărțit la numărul total de predicții.

accuracy

[1] 0.8777914



Graficul va afișa rata de fals pozitiv pe axa x și rata de adevărat pozitiv pe axa y.

```
Area under the curve: 0.8879
```

Rezultate si discutii:

Pentru a realiza o comparație și un clasament al modelelor prezentate anterior, am generat un grafic care suprapune curbele ROC. În acest grafic, putem observa că performanța unui model este cu atât mai bună cu cât suprafața descrisă de curba ROC este mai mare. Această suprafață este măsurată prin AUC. Astfel, prin analizarea curbelor ROC și a valorilor AUC

asociate fiecărui model, putem obține o perspectivă de ansamblu și putem clasifica modelele în funcție de performanța lor.