Proiect Fundamente de Big Data

Analiza calității laptelui

Echipă: Popoviciu Andreea & Samson Oana-Andreea

Email: andreea.popoviciu@stud.ubbcluj.ro, oana.samson@stud.ubbcluj.ro

Grupa: 5, An: 3

Introducere

Datorită procesului de evoluție continuu al conceptului de Machine Learning, astazi putem analiza și realiza preziceri pe seturi de date din diverse domeni precum medicină, piața imobiliarelor sau chiar agricultură și alimentație. Având în vedere contextul actual al unei societăți de consum în care clienții sunt din ce în ce mai informați în ceea ce privește calitatea alimentelor, producători trebuie să acorde mai multă atenție la menținerea produselor propri la un nivel cât mai ridicat. Problema propusă în discuție în cadrul studiului nostru vine în sprijinul consumatorilor care sunt preocupați de propria alimentație dar și al producătorilor dornici de a oferi produse de ceea mai înaltă calitate. Tema studiului constă în capacitatea de a prezice calitatea laptelui. Principalul aspect care contribuie la calitatea produselor lactate este laptele, acesta reprezintă o sursă de proteine și este o resursă importantă în dieta unei persoane deoarece contribuie la valoarea energetică.

Pentru producatori, îmbunătățirea calități laptelui va duce la produse lactate de înaltă calitate, aspect care va atrage atenția consumatorilor și astfel va contribui la creșterea profitului. De asemenea, un astfel de studiu poate ajuta clienți care își doresc să fie din ce în ce mai conștienți în ceea ce privesc produsele alimentare pe care le consumă.

Luând în calcul toate acestea, propunem următoarele întrebări la care ne dorim să găsim răspuns în cadrul studiului nostru:

- Care sunt factori care influențează cel mai mult calitatea laptelui?
- Cât de bine putem prezice calitatea laptelui?

Cunoașterea factorilor care influențează cel mai mult calitatea laptelui reprezintă un aspect important deoarece producători pot vedea cât de importante sunt anumite caracteristici și care sunt acele caracteristici pentru care ar trebui să acorde o atenție mai mare. Odată ce producători știu care sunt factori care influențează cel mai mult calitatea laptelui, aceștia vor fi mai conștienți și vor putea căuta modalități de a îmbunătăți aspectele respective. De asemenea, cunoașterea procentului de prezicere al calități laptelui este important deoarece producători trebuie să fie conștienți că modelul nu va avea preziceri de o corectitudine de 100%.

Setul de date

Setul de date este preluat de pe https://www.kaggle.com/datasets/cpluzshrijayan/milkquality. Acesta conține informați culese manual din diverse observați. Setul de date este format din 1060 de înregistrari. Fiecare înregistrare conține informați pentru 7 variabile independente reprezentate de pH (pH-ul laptelui), Temperature (temperatura laptelui în diverse stadi, de la producție până la consum), Taste (gustul), Odor (mirosul), Fat (procentul de grăsime), Turbidity (nivelul de opacitate) și Color (culoarea laptelui) și o variabilă dependentă Grade.

În tabelul de mai jos sunt reprezentate invervalele de valori pentru variabilele din setul nostru de date.

Variabile	Interval valori
pH	(3.0 – 9.5)
Temperature	(34 – 90)
Taste	(0, 1)
Odor	(0, 1)
Fat	(0, 1)
Turbidity	(0, 1)
Color	(240 - 255)
Grade	(low, high, medium)

Tabel 1

Atributele Taste, Odor, Fat și Turbidity vor arăta dacă laptele se încadrează în anumite condiți optime sau nu, astfel acestea vor avea valorile 1 pentru respectarea condițiilor și 0 pentru neconformarea cu acestea. Cele 4 atribute vor fi variabile nominale. Pentru coloanele pH, Temperature și Color sunt furnizate valorile numerice pentru acestea. Valoarea pentru pH a laptelui reprezintă o măsură a concentrației ionilor de hidrogen, acesta este utilizat pentru a măsura nivelul de aciditate sau alcalinitate. Pentru laptele proaspăt valoarea pH-ului este situate între 6.6 și 6.8. În ceea ce privește culoarea laptelui, conform RGB, valoarea 255 reprezintă nuanța cea mai intensă de alb, odată ce scade valoarea, scade și intensitatea de alb din culoare.

Aceste atribute prezintă interes în cadrul studiului nostru deoarece reprezintă câteva dintre cele mai importante caracteristici care pot influența calitatea laptelui. O valoare optimă a pH-ului este esențială pentru stabilitatea laptelui, acesta ajutând la prevenirea creșterii numărului de bacterii, de asemenea, pH-ul influențează la fermentarea și coagularea laptelui, utile la procesarea acestuia. Valori extreme de pH, foarte acide sau foarte alcaline, pot semnala probleme în ceea ce privește siguranța alimentară. Temperatura laptelui poate fi cauza creșterii și dezvoltarii bacteriilor, de asemenea aceasta poate duce la reducerea nutrienților, a vitaminelor și a enzimelor din lapte și poate afecta procesarea laptelui. Gustul poate fi un indicator important deoarece un gust proaspăt și dulceag va arăta că laptele este de calitate, în timp ce un gust anormal poate semnala probleme de contaminare cu diverse bacteri. Mirosul,

poate indica de asemenea dacă laptele este unul de calitate sau nu. Un alt factor important pentru a măsura calitatea laptelui este procentul de grasime, acesta poate influența textura, gustul și valoarea nutritive. De asemenea, nivelul de opacitate și culoarea laptelui constituie factori importanți în cee ace privește calitatea. Laptele proaspăt are de obicei o culoare albă sau ușor gălbuie și o turbiditate de nivel scăzut, fiind clar și transparent, orice variați ale acestor valori pot indica deteriorarea produsului și automat o calitate scăzută.

Variabila dependentă Grade va specifica calitatea laptelui, aceasta poate lua valorile Low pentru o calitate scăzută, Medium pentru o calitate medie și High pentru o calitate ridicată. Datorită faptului că variabila Grade conține valori nominale, variabila va fi una calitativă, iar datele poate fi împărțită în trei clase, setul nostru prezintă problema unei clasificări multiple.

În cadrul setului de date există 256 de instanțe ale clasei High, 429 instanțe ale clasei Low și 374 instanțe ale clasei Medium.

Deoarece, variabilele Taste, Odor, Fat, Turbidity și Grade sunt valori nominale le-am convertit în factori folosind funcția mutate().

În ceea ce privește curățarea datelor, nu am considerat că setul de date necesită modificări, toate atributele având valori numerice înafară de variabila dependentă și de asemenea, toate înregistrările având valori pentru toate atributele.

Rezultate și discuții

După încărcarea și procesarea setului de date pentru a realiza preziceri trebuie să învățăm un model. În cadrul acestui capitol, ne propunem să realizăm modele de clasificare folosind Naive Bayes, Arbori de decizie și Regersie Logistică. La final vom compara rezultatele de la toate metodele folosite.

Pentru a verifica nivelul de corelație între atributele setului de date vom face câte o matrice de corelație pentru cele trei clase ale variabilelei noastre țintă, astfel vom verifica dacă atributele sunt independente între ele. Datorită faptului că patru dintre atributele prezente în setul de date au valori nominale și au fost transformate în factori putem verifica nivelul de corelație doar pentru trei variabile, acestea fiind pH, Temperature și Color.

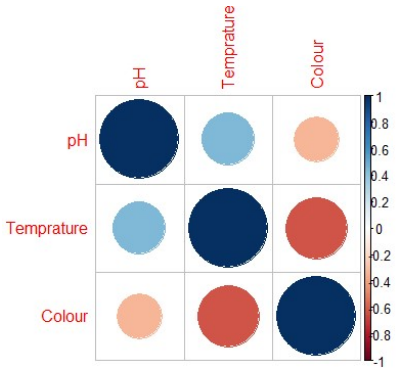

Figura 1.1

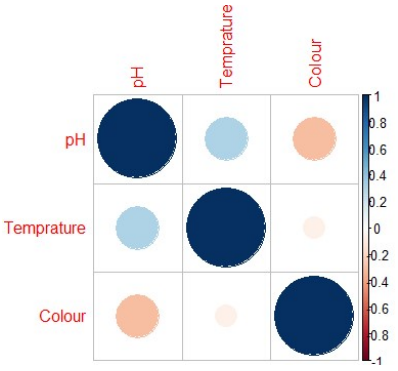

Figura 1.2

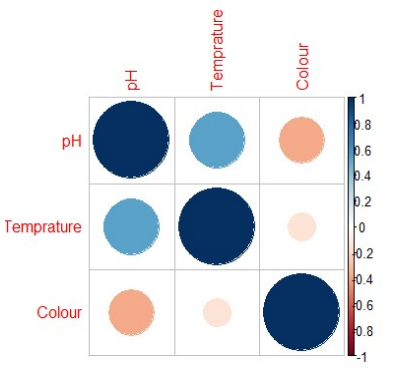

Figura 1.3

Nivelele de corelație sunt măsurate pe o scară de la [-1, 1], cel mai mare nivel de corelație fiind notat cu 1, iar cel mai mic fiind notat cu -1. Pentru toate cele trei situație există corelație maximă pe diagonală, unde se intersectează atributele și se măsuară nivelul de corelație cu ei înșiși. În cazul clasei High (Figura 1.1), se poate observa că există un nivel de corelație destul de ridicat între variabilele Temperature și pH, deoarece temperatura poate afecta creșterea activității enzimatice și apariția unor diverse bacteri, ceea ce va afecta pH-ul prin producerea de acid lactic și compuși acizi. De asemenea, se poate observa un nivel de corelație mediu spre scăzut între variabilele Color și pH, valoarea unui pH optim poate menține o culoare adecvată a laptelui în timp ce modificarea pH-ului, în cazul unor valori semnificative poate duce la variați de culoare. Totodată se poate observa un nivel de corelație scăzut pentru variabilele Color și Temperature, în general relația dintre cele două ar putea fi una indirectă sau infuențată de alți factori, culoarea fiind data de pigmenți naturali sau prezența unor impurități în timp ce temperatura afectează procese microbiologice și biochimice. Pentru clasa Low corelația dintre variabilele Teperature și pH are un nivel mai scăzut decât corelația celor două pentru clasa High. Corelația dintre Color și pH este asemănătoare cu ce de la clasa High iar corelația dintre Color și Temperature este mai ridicată dar prezintă o semnificație mai redusă decât cea de la clasa High. Pentru clasa Medium, comparativ cu clasa High, corelația între Temperature și pH prezintă un nivel mai ridicat, corelația între Color și pH este asemănătoare iar corelația între Color și Temperature prezintă un nivel puțin mai ridicat dar o semnificație mai scazută.

Pentru a învăța și testa un model este necesară împărțirea setului de date în două seturi diferite, de antrenament și de test. În cadrul studiului nostru am folosit o proporție de 70% pentru setul de antrenament și 30% pentru setul de test. Pentru a păstra proporțiile inițiale ale variabilelei dependente am folosit procesul de stratificare utilizând argumentul strata în funcție de variabila Grade.

Naive Bayes

Metoda Naive Bayes presupune că atributele sunt egale ca și importanță și sunt independente între ele. Aceasta calculează pentru fiecare instanță probabilitatea de apartenență la o anumită clasă și alege clasa de apartenență în funcție de cea mai mare probabilitate.

Începem prin a seta seed(123), pentru a menține datele reproductibile. După ce împărțim setul în date de antrenament și date de test observăm că avem pentru setul de antrenament 179 de instanțe ale clasei High, 300 pentru Low și 261 pentru Medium iar pentru setul de test avem 77 de instanțe ale clasei High, 129 ale clasei Low și 113 ale clasei Medium. Pentru ambele seturi se respectă proporțile variabilei dependente din setul inițial. Vom separa variabila țintă de varibelele independente și vom atribui pentru variabila x valorile variabilelor independente iar pentru y valorile variabilei dependente.

Stabilim o metoda de validare, în cazul nostru Cross Validation cu 10-folds folosind funcția trainControl(). Învățam modelul cu ajutorul funcției train() în care vom preciza variabilele independente, în cazul nostru x, variabila dependentă, reprezentată de y, metoda, în cazul nostru

nb, pentru Naive Bayes și specificăm metoda de validare, reprezentată de Cross Validation. Modelul nostru, datorită faptului că folosim Cross Validation cu 10-folds, va fi împărțit în 10 parți, va învăța pe 9 dintre ele și va testa pe al 10-lea, repetând acest process de 10 ori astfel încât toate instanțele să participe la procesul de teatare. În final, acesta va obține procente de acuratețe, prezente în tabelul de mai jos.

UseKernel	Accuracy
FALSE	0.9121592
TRUE	0.9243223

Tabel 2

În cazul în care modelul nu folosește kernel, adica pentru atributele numerice consideră variabilele în distribuție Gausiana, se obține o acuratețe de 91%. În cazul în care se folosește kernel obținem o acuratețe de 92%. Astfel putem vedea că performanța modelului va crește dacă folosim kernel cu 0.01. De asemenea, putem vedea posibilitatea folosiri metodei Laplace și ajustările realizate pentru kernel.

Pentru a verifica modelul afișam matricea de confuzie (Tabel 3).

	Reference		
Prediction	high	low	medium
high	72	3	9
low	0.0	116	0.0
medium	5	10	104
Accuracy (average)	0.9154		

Tabel 3

Pe baza matricei de confuzie, creată cu ceea mai bună variantă a modelului, ceea în care folosește kernel, putem observa că 72 de instanțe sunt prezise corect, ca aparținând clasei High, 116 sunt prezise corect, aparținând clasei Low, 104 sunt prezise corect ca aparținând clasei Medium. De asemenea, 5 sunt prezise greșit ca fiind High, acestea aparținând defapt clasei Medium, 3 și 10 sunt prezise greșit ca fiind Low și 9 sunt prezise greșit ca fiind Medium. Astfel rezultă o acuratețe de 91.54%. Ulterior am realizat predicți și curba ROC, unde am obținut o valoare pentru aria de sub curba (AUC) de 0.8443.

După care verificăm care combunație între parametri modelului este cea mai bună. Permitem modelului să folosească să folosească sau nu kernel, acordăm pentru Laplace valoarea 0.5, astfel pentru variabilele nominale va porni de la valoarea 0.5 în loc de 0, iar pentru kernel, aceștia vor fi ajustați cu secvențe de la 0 la 5 cu un pas de 1. Astfel vom avea 12 combinații posibile de modele, pe baza acestora vom alege modelul cel mai bun.

Folosim metoda train() la care adăugăm parametrul tuneGrid, cu ajutorul căruia pentru fiecare din cele 12 modele de mai sus se va realiza Cross Validation cu 10-folds. După cum putem observa din Figura 2, prezentă mai jos, putem vedea că modelul cel mai bun este cel în care se folosește kernel și o ajustare de 1, având o acuratețe de 91.48%. Toate modele din tabelul de mai jos au un fL de 0.5.

usekernel	adjust	Accuracy	Kappa
FALSE	0	0.9135651	0.8670159
FALSE	1	0.9135651	0.8670159
FALSE	2	0.9135651	0.8670159
FALSE	3	0.9135651	0.8670159
FALSE	4	0.9135651	0.8670159
FALSE	5	0.9135651	0.8670159
TRUE	0	NaN	NaN
TRUE	1	0.9148624	0.8695259
TRUE	2	0.8879069	0.8280369
TRUE	3	0.8906097	0.8320644
TRUE	4	0.8865556	0.8259606
TRUE	5	0.8825015	0.8198126

Figura 2

În continuare, vom prezice valori folosind cel mai bun model din cele generate mai sus, în variabila pred vom stoca valorile prezise iar în variabila predProb vom stoca probabilitățiile valorilor de a aparține la o anumită clasă. Astfel prezicerile alături de probabilități vor arăta asemănator cu cea din Tabel 4.

	high	low	medium
1	8.918076e-02	5.198938e-02	8.588299e-01

Tabel 4

În cazul acesta instanța va face parte din clasa Medium deoarece aceasta are probabilitatea cea mai mare

În cadrul matricei de confuzie realizată cu ajutorul predicțiilor și al valorilor reale, putem observa că pe setul de test avem o acuratețe de 91.54%, acuratețea se învârte între (0.8792, 0.9435), No Information Rate de 0.4044, aceasta arată că, în cazul în care am face o clasificare random, fără nici un model am obține o acuratețe de 40.44%, deoarece aceasta este distribuția pentru valorile claselor noastre. De asemenea, se poate observa P-Value care este mic, acuratețea fiind mai mare decât No Information Rate ipoteza nulă se respinge. Rata de specificitate pentru clasa High este 0.9351, pentru clasa Low este de 0.8992 iar pentru clasa Medium este 0.9204, iar rata de senzitivitate fiind 0.9504 pentru clasa High, 1.000 pentru clasa Low și 0.9272 pentru clasa Medium, aceestea fiind, de asemenea, valori bune.

Curba ROC reprezintă rata pentru true positives, instanțele clasificate corect reprezentate de senzitivitate, raportată la false positives, instanțele clasificate greșit, numite și specificitate. Apropierea de colțul din stânga sus determină o precizie bună a modelului, deoarece arată o rată de true positives mare și o rată mică de false positives. Vom genera curba ROC pentru valorile reale ale variabilei Grade din setul de test și probabilitățiile predicțiilor.

Datorită faptului că pentru setul nostru de date variabila dependentă este împărțită în trei clase, adică acesta prezintă problema unei clasificări multiple pentru a realiza curba ROC va trebui să folosim functia multiclass.roc(). Aceasta nu conține atribute predefinite pentru specificitate și senzitivitate, motiv pentru care în cadrul proiectului pentru a exprima performanța curbei ROC ne vom folosi de aria de sub curba (AUC). Astfel, cu cât va fi mai mare performanța acesteia cu atât AUC va avea o valoare mai mare

În cazul nostru, pentru metoda Naive Bayes, aria de sub curba ROC are o valoare de 0.8443, aceasta indicând o performanță medie spre mare a modelului.

În continuare, dorim să verificăm dacă aplicând parametri pentru cel mai bun model găsit, adică folosind kernel, un fL de 5 și ajustări pentru kernel de 1, vom obține o acuratețe mai mare fără a folosi Cross Validation. Astfel, folosind metode train() se va genera un singur model pe baza parametrilor găsiți mai sus. Din matricea de confuzie pentru acest model putem observa că valoarea pentru acuratețe de 0.9154, aceasta fiind mai mică decât acuratețea pentru celelalte modele.

O reprezentare grafică a celor 12 modele create pentru a găsi parametri cu cele mai bune valori este prezentă în Figura 3. Astfel putem vedea că modelele care nu folosesc kernel, adică consideră o distribuție Gausiana pentru atributele numerice, valorile sunt în jur de 0.914. Pentru modelele care folosesc kernel acuratețea crește pentru ajustări de 1 și scade pentru valorile 2, 3, 4, 5. De asemenea, acuratețea crește puține pentru ajustări cu valoarea 3 față de ajustări cu valoarea 2.

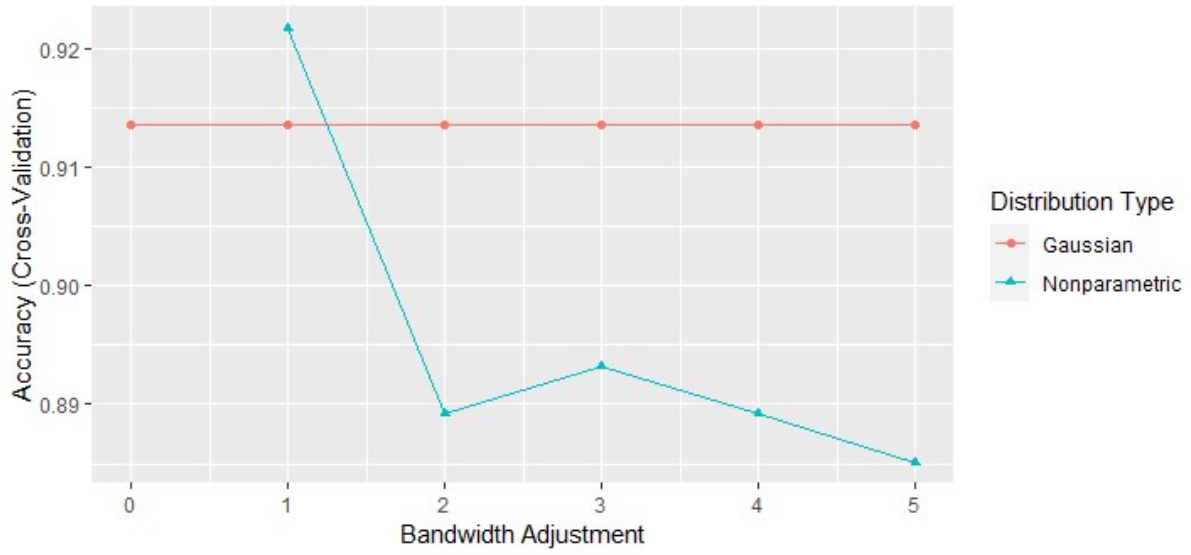

Figura 3

În final dorim să optimizat modelul în funcție de curba ROC, pentru a vedea dacă modelul cu cea mai bună arie de sub curba ROC (AUC) va avea ceea mai mare acuratețe.

usekernel	adjust	logLoss	AUC	prAUC	Accuracy	Kappa	Mean_F1
FALSE	0	0.1852225	0.9929794	0.6789280	0.9134925	0.8669368	0.9007390
FALSE	1	0.1852225	0.9929794	0.6789280	0.9134925	0.8669368	0.9007390
FALSE	2	0.1852225	0.9929794	0.6789280	0.9134925	0.8669368	0.9007390
FALSE	3	0.1852225	0.9929794	0.6789280	0.9134925	0.8669368	0.9007390
FALSE	4	0.1852225	0.9929794	0.6789280	0.9134925	0.8669368	0.9007390
FALSE	5	0.1852225	0.9929794	0.6789280	0.9134925	0.8669368	0.9007390
TRUE	0	NaN	NaN	NaN	NaN	NaN	NaN
TRUE	1	0.2159330	0.9906731	0.9131447	0.9243219	0.8841873	0.9188967
TRUE	2	0.2457424	0.9912359	0.9142237	0.8864825	0.8257549	0.8780077
TRUE	3	0.2796834	0.9865881	0.8966578	0.8918879	0.8340430	0.8832248
TRUE	4	0.3015481	0.9836885	0.8923546	0.8878339	0.8278904	0.8798606
TRUE	5	0.3184230	0.9808418	0.8870814	0.8837798	0.8217059	0.8765294

Figura 4

După cum putem vedea în Figura 4, în continuare modelul cu cea mai bună acuratețe este în condițile în care folosește kernel și are ajustări de 1.

Realizam predicțiile și probabilitățiile predicților și afișăm matricea de confuzie unde obținem o acuratețe de 89.97%, mai mică decât în cazul anterior în care optimizăm modelul în funcție de parametri.

Din punctul nostru de vedere cea mai bună variantă este al doilea model unde am folosit Cross Validation cu 10-folds kernel și o ajustare, de 1, având o acuratețe de 91.48% și aria de sub curba (AUC) de 0.8443.

Arbori de decizie pentru clasificare

Abordarea Cart este utilizată în partiționarea datelor în subseturi mai mici pe baza testelor, iar toate instanțele dintr-un subgrup sunt caracterizate prin aceeași valoare de predicție. Fiecare subset frunză, potrivește clasa care apare de cele mai multe ori între observațiile din subset, iar pentru a minimiza rata de eroare se relizează selecția testului din noduri.

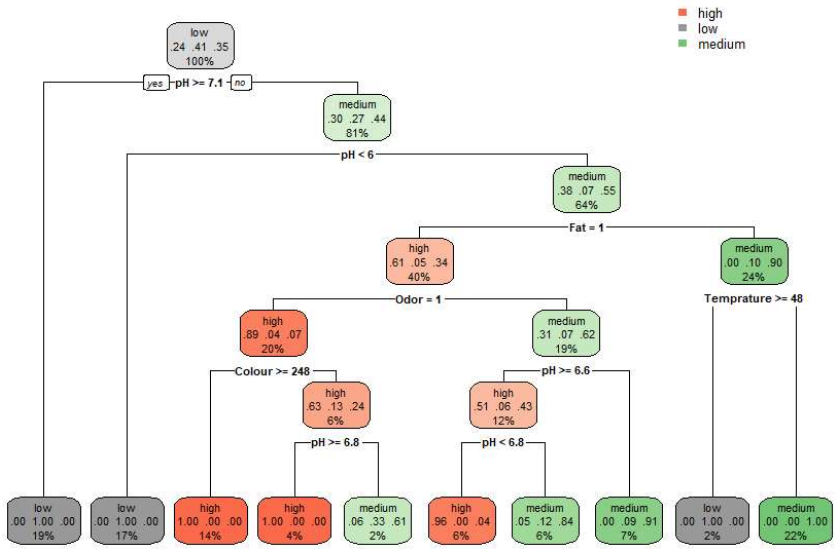

Figura 5

În vârful arborelui avem 740 de instanțe și 440 de erori aparținând clasei (low). Pe nodul 2 avem atributul pH, care este primul nod copil ar rădăcinei, unde avem un număr de 141 de instanțe care au fost luate în considerare pentru a satisface condiția "pH >=7.1", fiind asociate clasei (low)[1]. Pe nodul 3 avem un număr de 599 de instanțe care nu îndeplinesc condiția anterioară, aparținând clasei (midium), iar nodurile 6 și 7 care au tot ca și atribut pH-ul fiind caracterizate de valori mai mici decât 6 și respectiv mai mari sau egale cu 6. În continuare avem nodurile 17 și 28 care se împart după atributele Fat și Odor, unde avem un număr de 294 și respectiv 151 de instanțe care sunt luate în considerare pentru că satisface condiția "=1", fiind asociate, ambele atribute, clasei (high). Urmează Color care este caracterizată de o valoare mai mare sau egala 247.5, cărora li se asociază clasa (high) și respectiv cu o valoare mai mică de 247.5 fiind asociată aceeași clase.

[1] Simbolul "*" indică nodul terminal în arbore

Astfel, arborele de decizie prezintă în vârful său o clasă majoritară, împărțind datele în funcție de nivelul pH. Dacă nivelul este mai mic de 7.1 avem două ramuri, pentru instanțele cu un nod mai mic de 6 sunt clasificate în clasa (low) fără alte despărțiri suplimentare și cele cu noduri mai mari sau egale cu 6, fiind clasificate în clasa (medium). În continuare, avem o ramificație după atributele Fat și Odor, ambele fiind egale cu 1 si având aceeași clasificare după clasa (high). Urmează încă o ramificație în arbore pentru atributul Color care este împărțit pe două ramuri, pentru instanțele cu un nod mai mare de 247.5 și cele cu noduri mai mici de 247.5, ambele fiind clasificate în clasa (high). Astfel, cu ajutorul arborelui putem să vedem care sunt factorii cei mai importanți. În tabelul de mai jos sunt datele menționate anterior (Tabel 5) și o imagine cu arborele (Figura 4)

Node	Split	n	Loss	Class	Probability
1	Root	740	440	low	0.24189189, 0.40540541, 0.35270270
2	pH>= 7.1	141	0	low	0.00000000, 1.00000000, 0.00000000
3	pH< 7.1	599	338	medium	0.29883139, 0.26544240, 0.43572621
6	pH< 6	125	0	low	0.00000000, 1.00000000, 0.00000000
7	pH>= 6	474	213	medium	0.37763713, 0.07172996, 0.55063291
14	Fat=1	294	115	high	0.60884354, 0.05442177, 0.33673469
28	Odor=1	151	17	high	0.88741722, 0.03973510, 0.07284768
56	Color>= 247.5	105	0	high	1.00000000, 0.00000000, 0.00000000
57	Color< 247.5	46	17	high	0.63043478, 0.13043478, 0.23913043

Tabel 5

Cu ajutorul parametrului “cp=0 ” arborele nu va fi taiat deloc și se va ajusta perfect pe setul nostru de date, obținând acuratețea 94.98%. Totodată folosind summary() vizualizăm o descriere a arborelui unde este inclusă și importanța variabilelor. Cele mai importante sunt pH, Temperature și Fat, iar cele cu o importanță mai scăzută sunt Odor, Color, Turbidity și Taste (Tabel 6).

Variabile Importante						
pH	**Temperature**	**Fat**	**Odor**	**Color**	**Turbidity**	**Teste**
44	22	11	7	6	5	4

Tabel 6

De asemenea, am vrut să testăm acuratețea pentru un arbore tăiat după valoarea parametrului

“cp = 0.04” și am obținut acuratețea de 92.48%.

Pentru evaluarea arborilor de decizie am utilizat Indexul Gini, care măsoară varianța totală pe cele k clase și Entropia, care va lua valorile cele mai mici dacă un nod este pur. În urma testării celor doi indici am observat că acuratețea este una mult prea mare, modelul fiind supraestimat și poate să ducă la overfitting (Tabel 7).

Accuracy	
Entropie	0.9781
Gini	0.953

Tabel 7

Bagging

Metoda de agregare a deciziilor, numită Bagging implică creșterea eșantionului Bootstrap din set și ajustează arborele de decizie. Pentru a se obține valoarea de predicție, se face o medie a predicțiilor individuale obținute pentru fiecare arbore.

Ne vom ajuta de procedura coob pentru a putea estimam performanța arborilor pe instanțele de date care sunt excluse în fiecare eșantion Boostrap. Prin evaluarea performanței pe instanțele OOB (Out-of-bag), putem să determinăm eraorea de clasificare, în cazul nostru valoarea obținută este 0.0014, fiind atât de mică putem să spunem că modelul nostru a obținut o performanță foarte bună, iar pentru construirea modelului am uitilizat 32 de bags, după cum putem să observăm linia graficului nu mai scade deci putem să considerăm că rata de eroare s-a stabilizat (Figura 6).

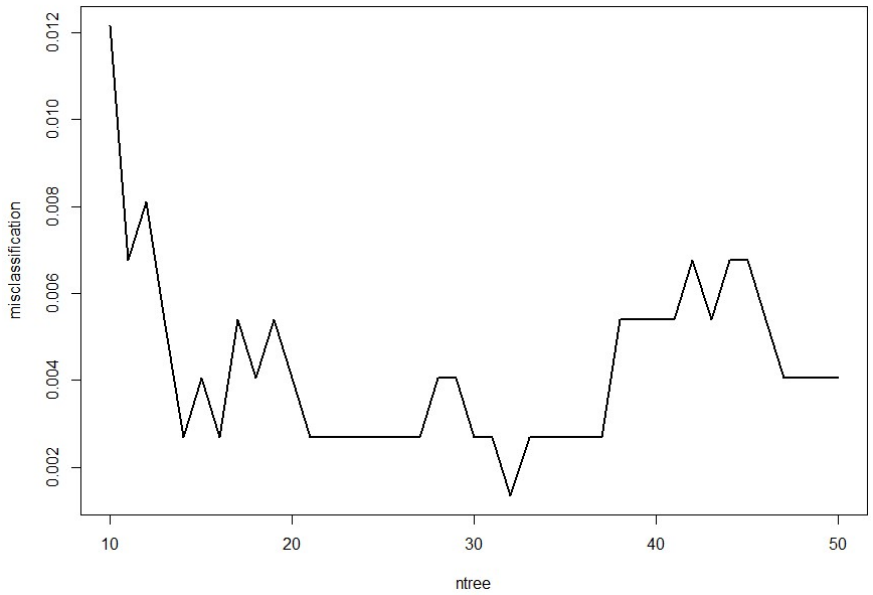

Figura 6

Random Forest

Random Forest combină arborii de decizie pentru a obține rezultate mai precise și extrage un eșantion de Bootstrap, pe care se învață un arbore Cart.

Modelul pe care l-am folosit la Random Forest este de tip clasificare și are 500 de arbori și la fiecare split al arborilor avem 2 variabile. Erorea de clasificare OOB (Out-of-Bag) este de

0.14% ceea ce ne indică că modelul are o performanță bună pe setul de date de antrenament. Matricea de confuzie prezintă rezultatele celor trei clase (low, medium și high) de aici putem să obervăm că în clasa (high) avem 179 de clasificări corecte din 179, în clasa (low) avem 299 de clasificări corecte din 300 și în clasa (medium) avem 261 de clasificări corecte din 261, iar în coloana "class.error" erorile sunt extrem de mici sau chiar 0 (Tabel 8)

	High	Low	Medium	Class.error
High	179	0	0	0.000000000
Low	1	299	0	0.003333333
Medium	0	0	261	0.000000000

Tabel 8

Comparația metodelor folosite la Arborii de Clasificare

Comparând rezultatele obținute prin aplicarea celor cinci metode asupra setului de date observăm că atât metoda Bagging cât și metoda Random Forest au cele mai bune rezultate. Acuratețea fiind de 99.97%, o cauză ar putea să fie și marimea arborelui care este destul de scăzută, având un număr mic de atribute, iar acest lucru ne-a determinat să nu mai facem și o randomizare, deoarece rezultatele se pot pierde. Specificitatea celor doua modele este de 100% ceea ce înseamă că modelul nu a identificat niciun caz negativ sau greșit în setul nostru de date. După, putem să vedem că atât metoda Entropie cât și Indexul Gini au o acuratețe un pic mai scăzută decăt Bagging și Random Forest, fiind de 97.81% și respectiv 95.30%, ambele rezultate demostrează ca metodelele sunt și ele la fel de eficiente, iar la Arbore de decizie putem să vedem că acuratețea este de 94.98%, mult mai mică față de celelalte patru metode. De asemena, pentru Arborele Tăiat am obținut o acuratețe de 92.48%, cea mai mica dintre cele 5 valori. P-value este foarte mic pentru toate metodele menționate de mai jos, ceea ce sugerează o încredere ridicată în rezultatele pe care le-am obținut (Tabel 9).

	Accuracy	P-Value
Arbore Simplu	0.9498	< 2.2e-16
Arbore Taiat	0.9248	< 2.2e-16
Entropie	0.9781	< 2.2e-16
Gini	0.953	<2e-16
Bagging	0.9937	< 2.2e-16
Random Forest	0.9937	< 2.2e-16

Tabel 9

Am ales să facem curba ROC pentru Arborele simplu, Entropie și Gini, am exclus metodele Bagging și Random Forest deoarece au obținut o acuratețe prea mare iar modelul poate fi supraestimat. Astfel, folosind funcția multiclass.roc() am obținut aria de sub curbă (AUC) conform tabelului de mai jos.

	AUC
Arbore Simplu	0.9863
Arbore Tăiat	0.8455
Entropie	0.9913
Gini	0.9913

Tabel 10

Din punctul nostru de vedere, cea mai optimă variantă rămâne Arborele Tăiat deoarece chiar dacă are acuratețea cea mai mică pare mai credibil și este mai predispus la overfitting.

Regresie Logistică

Pentru regresia logistică este necesar să avem două clase țintă, astfel folosind regresia vom obține o probabilitate între 0 și 1, clasa de care aparține instanța va fi data în funcție de un prag ales de noi, de exemplu 0.5, valorile care vor fi sub prag vor aparține primei clase iar valorile care sunt peste prag vor aparține celei de-a doua clasa.

Deoarece setul nostrum de date prezintă problema unei clasificări multiple vom folosi funcția vglm() din pachetul VGAM pentru a realiza regresia logistică

Pentru a vizualiza mai bine datele vom afișa câteva grafice.

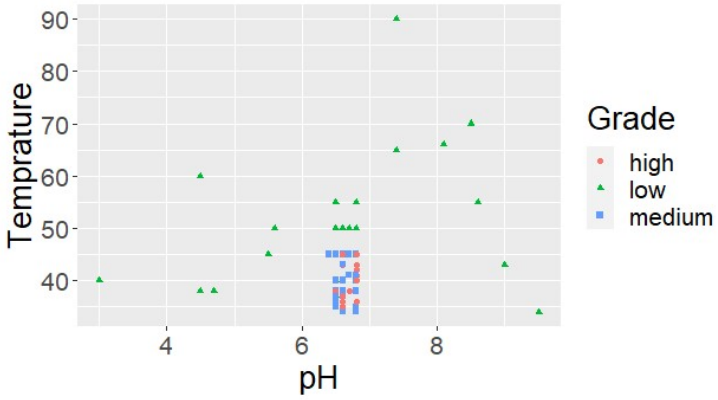

Figura 7

În Figura 7 am realizat un grafic între termperatură și pH în funcție de clasa de apartenență a instanțelor. Aici putem vedea ca instanțele care aparțin clasei High și Medium au o valoare pentru pH situată aproximativ între 6.4 și 6.9 iar pentru temperatură o valoare situată între 35 și 45 de grade Fahrenheit. Pentru orice variați ale oricărei variabile instanța aparține clasei Low.

În Figura 8.1 și Figura 8.2 putem observa mai detaliat cum este influențată calitatea laptelui de pH și temperatură. În prima figură putem observa că balanța instanțelor din clasa High are valori pentru pH aproximativ între 6.5 și 6.9 având media pe la 6.7. Pentru clasa Low valorile pentru pH pot varia între 3 și 9.3, majoritatea valorilor fiind între 4.8 și 8.6 media fiind de 6.8. Pentru clasa Medium valoarile pot varia între 6.5 și 6.9, valoarea medie fiind de 6.6. În cea de-a doua figură putem vedea că balanța pentru clasa High are valorile pentru temperatură situate între 35 și 45 de grade având mediua în jurul valori de 40 de grade. Pentru clasa Low valoarile variază între 34 și 70 de grade având o extremitate de 90 de grade, majoritatea valorilor fiind între 40 și 55 de grade cu media de 45 grade. Pentru clasa Medium valorile variază între 34 și 45 de grade având o medie de aproximativ 37 grade.

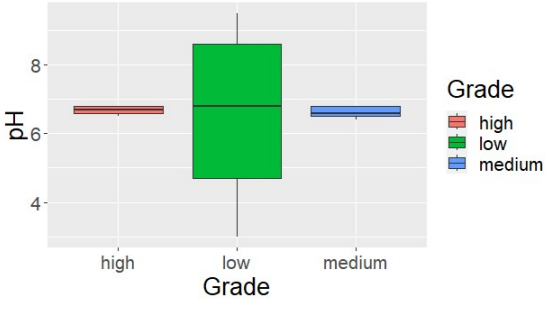

Figura 8.1

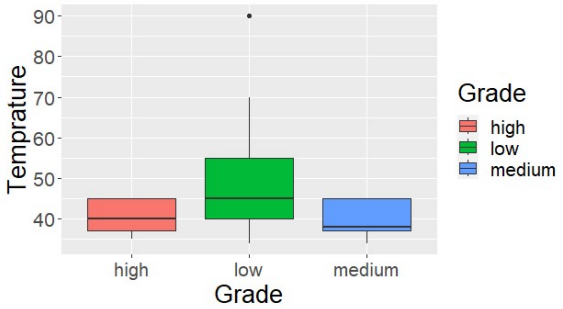

Figura 8.2

Începem prin a seta un seed(123) pentru a menține datele reproductibile. Vom împărți setul de date în 70% date de antrenament și 30% date de test. Pentru a creea modelul de regresie logistică multiplă o să folosim funcția vglm(), în cadrul căreia vom solicita ca modelul să fie generat pentru Grade în funcție de restul variabilelor, vom specifica setul de date iar pentru parametrul family vom specifica valoarea multinomial.

Funcția summary() va descrie modelul. Pentru regresia multinomială este necesar să avem estimări separate pentru fiecare nivel al categoriilor. Astfel modelul va lua o clasă de referință a variabilei noastre țintă pentru a compara celelalte clase, prin urmare clasa de referință va reprezenta nivelul de bază cu care se vor compara celelalte niveluri. În cazul modelului nostru este ales nivelul 3 al variabilei Grade, clasa Medium, ca și nivel de referință. De exemplu, coeficientul pentru variabila pH:1 va estima cât de mult influențează pH-ul probabilitatea variabilei Grade de a aparține clasei Medium iar coeficientul pentru pH:2 va estima influența variabilei pH asupra probabilități de apartenență a variabilei Grade la o altă clasă în afară de Medium.

În Tabel 11 putem observa că toate valorile din coloana Estimate au valor pozitive înafară de nivelul 2 pentru pH, pentru valorile pozitive putem spune că la o creștere a valori variabilelor vom avea o creștere a probabilități ca variabila Grade să aparțină unei clase comparativ cu clasa Medium deoarece vom avea o creștere a lui log-odds, în schimb pentru valorile negative vom avea o scădere a probabilități . De exemplu, pentru pH:1, o creștere a valori pH cu o unitate va determina creșterea probabilități cu 0.01439 ca variabila Grade să aparțină clasei Medium, în schimb, pentru pH:2 observăm că o creștere a valori pH va determina o scădere a probabilități cu 0.72812 ca variabila Grade să aparțină unei alte clase diferită de Medium, adică High sau Low. Acest lucru nu este foarte valid deoarece valorile variabilei pH pentru instanțele din clasa High sau Medium sunt în intervalul 6.5 și 6.9. De asemenea, putem observa că toți coeficienți, în afară de pH:1, au un grad de importanță mare și au valori pentru p-value mici sau foarte mici. În schimb, pH:1 prezintă o valoare mare pentru p-value ceea ce arată că o creștere a valori de pH nu va influența foarte mult apartenența la clasa Medium ceea ce înseamnă că ipoteza nula se acceptă. Coeficientul (Intercept):1 indică estimarea log-odds și probabilitatea de apartenență la clasa Medium iar (Intercept):2 indică estimarea log-odds și prin urmare posibilitatea de apartenență la cele două clase diferite de Medium, adică High sau Low. Deoarece (Intercept):1 are valori negative arată că valoarea de log-odds pentru clasa de referință, cea de nivel 3, este mai mici decât valorile de log-odds pentru clasele High sau Low, prin urmare probabilitatea de apartenență la clasa Medium este mai mică.

	Estimate	Std. Error	Z value	Pr(>|z|)	Significance
(Intercept):1	-208.27058	23.85721	-8.730	< 2e-16	***
(Intercept):2	-234.20823	23.42903	-9.996	< 2e-16	***
pH:1	0.01439	0.16323	0.088	0.929732	
pH:2	-0.72812	0.13755	-5.294	1.20e-07	***
Temperature:1	0.18515	0.05365	3.451	0.000558	***
Temperature:2	0.60595	0.05234	11.577	< 2e-16	***
Taste:1	3.62566	0.41420	8.753	< 2e-16	***
Taste:2	5.36951	0.47493	11.306	< 2e-16	***
Odor:1	5.87921	0.57548	10.216	< 2e-16	***
Odor:2	3.38190	0.54444	6.212	5.24e-10	***
Fat:1	10.81103	1.20663	8.960	< 2e-16	***
Fat:2	2.52383	0.41070	6.145	7.99e-10	***
Turbidity:1	3.22058	0.53058	6.070	1.28e-09	***
Turbidity:2	5.21341	0.48867	10.669	< 2e-16	***
Color:1	0.73570	0.08599	8.555	< 2e-16	***
Color:2	0.81560	0.08439	9.665	< 2e-16	***

Tabel 11

După ce antrenăm modelul vom realiza predicți. Pentru regresie logistică se va stabili un prag în funcțiie de care se vor prezice valorile. Deoarece modelul nostru este unul pentru clasificare multiplă sunt necesare anumite procesări pentru a aplica pragul pentru regresie. Pentru a obține o acuratețe cât mai mare am încercat diferite valori pentru pragul regresiei.

Folosind un prag de 0.2 am obținut următoarea matrice de confuzie (Tabel 11.1), realizată pe baza predicțiilor generate și a setului de date de test care conține în total 319 instanțe.

	Reference		
Prediction	high	low	medium
high	68	31	15
low	5	97	8
medium	4	1	90
Accuracy (average)	0.7994		

Tabel 11.1

Aici putem observa că 68 de instanțe au fost prezise corect, acestea aparținând clasei High, 97 instanțe au fost prezise corect pentru clasa Low și 90 de instanțe au fost prezise corect pentru clasa Medium. De asemenea 5 și 4 instanțe au fost prezise greșit pentru clasa High, 31 și 1 instanțe au fost prezise greșit pentru clasa Low iar 15 și 8 instanțe au fost prezise greșit pentru clasa Medium. Astfel modelul nostru a obținut o acuratețe de 0.7994.

Pentru un prag de 0.3 matricea de confuzie (Tabel 11.2) va indica o acuratețe de 0.837. Aici putem observa că 63 de instanțe au fost prezise corect pentru clasa High, 109 instanțe au fost prezise corect pentru clasa Low și 90 de instanțe au fost prezise corect prentru clasa Medium. Astfel din cele 314 valori prezente în setul nostrum de test 52 au fost prezise greșit.

	Reference		
Prediction	high	low	medium
high	63	19	15
low	5	109	8
medium	4	1	90
Accuracy (average)	0.837		

Tabel 11.2

De asemena, pentru un prag de 0.5 vom obține matricea de confuzie prezentă în Tabel 11.3 și o acuratețe de 0.8213.

	Reference		
Prediction	high	low	medium
high	59	10	15
low	14	113	8
medium	4	6	90
Accuracy (average)	0.8213		

Tabel 11.3

Astfel după cum putem observa pragul care ne ajută să obținem cea mai mare acuratețe este de 0.3.

În final am realizat curba ROC, folosind funcția multiclass.roc(), pentru care aria de sub curba (AUC) are o valoare de 0.9392. De asemenea, am vrut să verificăm care sunt variabilele care influențează cel mai mult calitatea laptelui. Pentru a face acest lucru am realizat două de modele de regresie, unul în care variabila dependentă este influențată de cele mai importante variabile (pH, Termperature, Fat), conform arborelui de decizie și unul în care variabile dependetă este influențată de variabilele independente cu o importanță mai scăzută (Taste, Odor, Turbidity, Color).

	Reference		
Prediction	high	low	medium
high	39	28	35
low	37	88	10
medium	1	13	68
Accuracy (average)	0.6113		

Tabel 12.1

	Reference		
Prediction	high	low	medium
high	28	23	12
low	35	74	11
medium	14	32	90
Accuracy (average)	0.6019		

Tabel 12.2

În Tabel 12.1 avem matricea de confuzie pentru regresia realizată în funcție de variabilele importante iar în Tabel 12.2 avem matricea de confuzie pentru variabilele mai puțin importante. După cum putem observa folosind variabilele cu o importanța mai mare a obținut o acuratețe de 0.6113, mai mare decât în cazul modelului realizat în funcție de variabilele cu o importanță mai scăzută, de 0.6019.

Compararea rezultatelor obținute

În urma studiului realizat pe baza setului de date am ales câte un model pentru fiecare dintre cele trei metode în parte.

	AUC	Acurracy	95% CI	NIR	P-Value
Naive Bayes	0.8467	0.9243	(0.8792, 0.9435)	0.4044	< 2.2e-16
Arbori de decizie	0.8455	0.9248	(0.8901, 0.9512)	0.4044	< 2.2e-16
Regresie Logistică	0.9392	0.8213	(0.7748, 0.8618)	0.4044	< 2e-16

Tabel 13

În Tabel 13 avem valorile pentru aria de sub curba ROC (AUC) obținută folosind funcția multiclass.roc() și valorile matricei de confuzie realizată pe baza setului de test și predicțiilor realizate cu ajutorul modelului. Astfel putem observa că pentru modelul Naive Bayes avem o valoare pentru AUC de 0.8467, o acuratețe de 92.43%, acuratețea se învârte între (0.8792, 0.9435). Pentru Arbori de decizie avem o valoare pentru AUC de 0.8455, o acuratețe de 92.48%, acuratețea se învârte între (0.8901, 0.9512). Pentru Regresie Logistică avem o valoare pentru AUC de 0.9392, o acuratețe 82.13%, acuratețea se învârte între (0.7748, 0.8618). Toate cele trei metode au o valoare pentru No Information Rate (NIR) de 0.4044, aceasta arată că, în cazul în care am face o clasificare random, fără nici un model am obține o acuratețe de 40.44%, deoarece aceasta este distribuția pentru valorile claselor noastre. De

asemenea, P-value este foarte mic pentru toate metodele menționate, ceea ce sugerează o încredere ridicată în rezultatele pe care le-am obținut.

În urma analizei noastre pe setul de date considerăm că modelul cel mai bun este reprezentat de Arborii de decizie deoarece a obținut o acuratețe de 92.48% și răspunde cel mai bine la întrebările de cercetare.

Prima întrebare făcea referire la factori care influențează cel mai mult calitatea laptelui. Conform descrieri arborelui ales am putut să clasificăm variabilele independente în funcție de importanță. Astfel, cele mai importante variabile sunt pH, Temperature și Fat, iar cele cu o importanță mai scăzută sunt Odor, Color, Turbidity și Taste. Acest aspect este adevărat deoarece, pH, Temperatura și procentul de grăsime sunt esențiale în ceea ce privește calitatea laptelui.

Valoarea pH-ului influențează fermentarea și coagularea laptelui și ajută la prevenirea creșterii numărului de bacterii. De asemenea, temperatura laptelui poate fi cauza creșterii și dezvoltării bacteriilor și poate duce la reducerea nutrienților, a vitaminelor și enzimelor, iar procentul de grăsime este un alt factor important acesta influențând textura, gustul și valoarea nutritivă. În ceea ce privește culoarea, gustul, mirosul și nivelul de opacitate, acestea sunt predispuse să fie influențate de primele trei variabile independente.

A doua întrebare de cercetare făcea referire la cât de bine putem prezice calitatea laptelui, astfel am ajuns la concluzia că putem prezice acest lucru cu o acuratețe de 92.48%.

Concluzia

În concluzie, cel mai bun model pentru setul nostru de date, care prezintă problema unei clasificări multiple, este reprezentat de Arborii de Decizie. Am testat mai multe variante în cadrul capitolului “Rezultate și discuții” și am considerat că modelul pentru un arbore tăiat după valoarea parametrului “cp = 0.04” ar obține cele mai bune rezultate luând în considerare că setul nostru are o dimensiune redusă de date, totodată acesta având doar trei variabile numerice și patru variabile nominale, exceptând variabila dependentă (Grade). Astfel, am aflat că factorii care influențează cel mai mult calitatea laptelui sunt pH-ul, temperatura și procentul de grăsime și am ajuns la performanța de a prezice calitatea laptelui cu o acuratețe de 92.48%.