Documentatie Invatare Automata

~ Zoo Animal Classification ~

Studenta:Brânzea Ana-Maria

Grupa 40317A

CUPRINS

Analiza bazei de date 3
Modelul Perceptron 4
Modelul Naive Bayes 7
Modelul Knearest Neighbors 8
Modelul Support Vector Machine 10
Modelul arbori de decizie 12
Concluzii 15

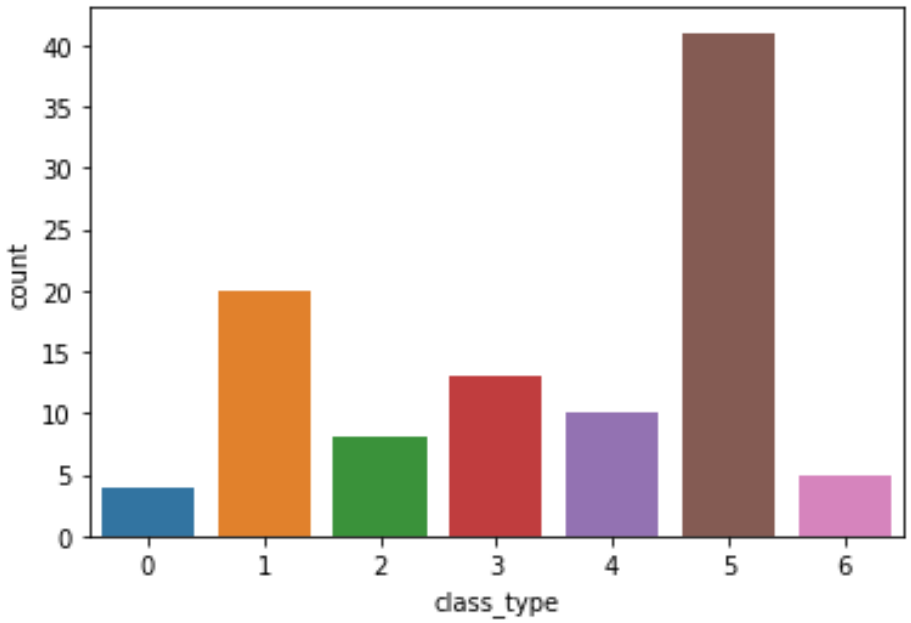
Analiza bazei de date

Baza de date utilizată este Zoo Animal Classification disponibilă pe platforma Kaggle şi este formată din 101 înregistrări. Descrierea parametrilor este următoarea:

- animal_name: Numele animalului.
- hair: Prezenţa părului (1 - Da, 0 - Nu).
- feathers: Prezenţa penei (1 - Da, 0 - Nu).
- eggs: Tipul de reproducere (1 - Ouă, 0 - Altele).
- milk: Secretare de lapte (1 - Da, 0 - Nu).
- airborne: Capacitatea de a zbura (1 - Da, 0 - Nu).
- aquatic: Trăieşte în apă (1 - Da, 0 - Nu).
- predator: Este un prădător (1 - Da, 0 - Nu).
- toothed: Are dinţi (1 - Da, 0 - Nu).
- backbone: Are coloană vertebrală (1 - Da, 0 - Nu).
- breathes: Respiră (1 - Da, 0 - Nu).
- venomous: Este veninos (1 - Da, 0 - Nu).
- fins: Prezenţa înotătoarelor (1 - Da, 0 - Nu).
- legs: Numărul de picioare (valori numerice- 0,2,4,5,6,8).
- tail: Prezenţa cozii (1 - Da, 0 - Nu).
- domestic: Este domestic (1 - Da, 0 - Nu).
- catsize: Dimensiunea similară cu cea a unei pisici (1 - Da, 0 - Nu).
- class_type: Tipul de clasă (nominal- [1,7]): Mammal, Bird, Fish, Amphibian, Reptile, Bug, Invertebrate.

Baza de date nu conţine valori lipsă, motiv pentru care nu s-au realizat modificări asupra datelor iniţiale. Distribuţia datelor pentru cele 7 clase este următoarea:

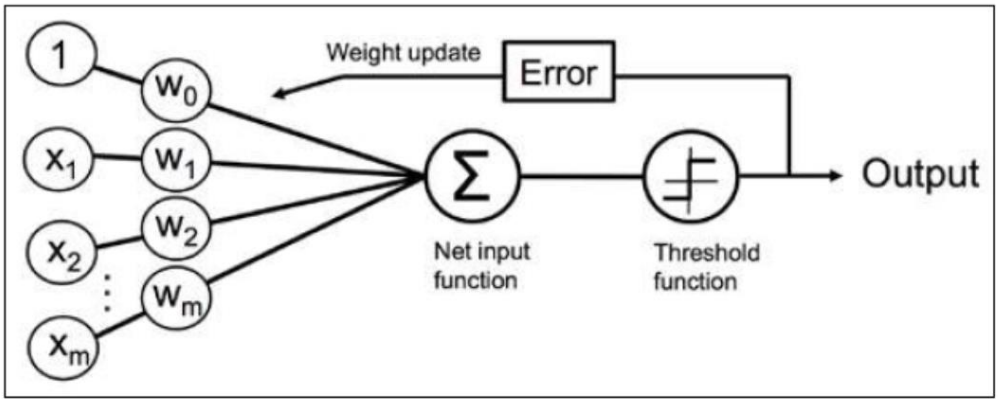
- Clasa 0 (Amphibian) : 4 inregistrari
- Clasa 1 (Bird): 20 de inregistrari
- Clasa 2 (Bug) : 8 inregistrari
- Clasa 3(Fish): 13 inregistrari
- Clasa 4(Invertebrate):10 inregistrari
- Clasa 5(Mammal):41 de inregistrari
- Clasa 6(Reptile): 5 inregistrari



Cele cinci modele utilizate pe această bază de date sunt: Perceptronul, Clasificatorul Naive Bayes, Clasificatorul K-Nearest Neighbors, Maşina de Vectori Suport (SVM) şi Arborii de Decizie.

Modelul Perceptron

Fiecare informaţie are o pondere asociată. În interiorul neuronului, se calculează suma ponderilor, iar pe baza unei probabilităţi, se determină care rezultat este mai apropiat de cel aşteptat. După efectuarea acestei comparaţii şi stabilirii legăturii din neuron care ar putea fi întărită, se modifică ponderea şi se recalculează eroarea.



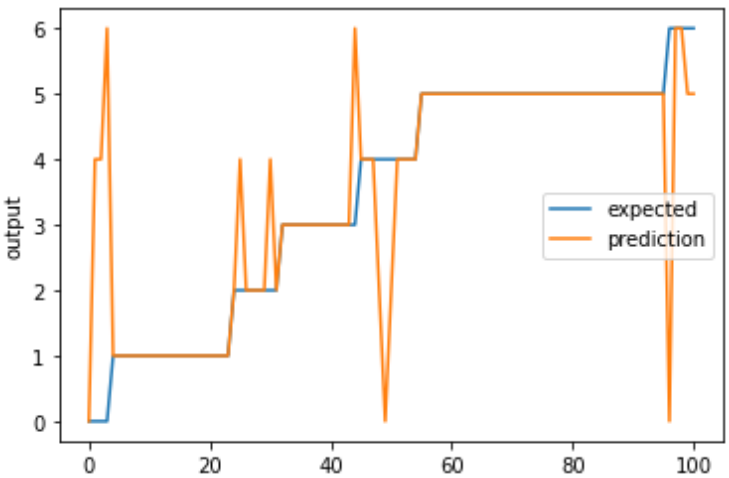
Pentru modelul Perceptron, parametrii utilizaţi sunt:

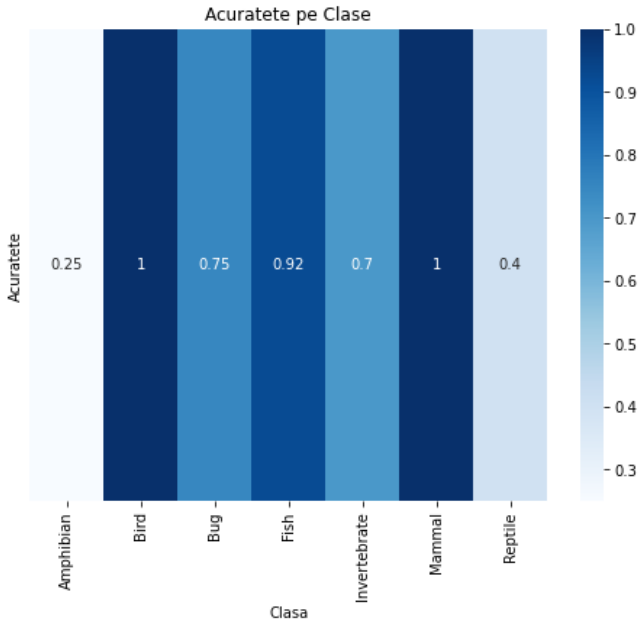
- Parametrul k- numarul de folduri folosite
- Eta0- rata cu care se ajustează ponderile modelului în timpul antrenamentului
- Max_iter- numărul maxim de iteraţii sau paşi de antrenament pe care algoritmul de învăţare le va efectua în timpul antrenamentului
- Early_stopping- un indicator care controlează dacă antrenamentul ar trebui să se oprească devreme în funcţie de anumite criterii (ia valorile True/False)

Experimentele modelului Perceptron:

Nume experiment	Parametrul k	Rata de invatare(eta 0)	Epoci (max_iter)	Valoare early_stopping	Acuratete medie	Matricea de confuzie
Exp1	5	0.05	40	true	0.88	[[2 0 0 0 0 0 2] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [3 2 1 0 4 0 0] [0 0 0 0 0 41 0] [2 0 0 0 0 2 1]]
Exp2	5	0.02	60	true	0.88	[[2 0 0 0 0 0 2 0] [0 20 0 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [1 2 1 0 6 0 0] [0 0 0 0 0 41 0] [4 0 0 0 0 1 0]]
Exp3	5	0.01	60	false	0.90	[[2 0 0 0 0 0 2 0] [0 20 0 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [1 2 1 0 6 0 0] [0 0 0 0 0 41 0] [4 0 0 0 0 1 0]]
Exp4	5	0.01	60	true	0.88	[[2 0 0 0 0 0 2] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [3 2 1 0 4 0 0] [0 0 0 0 0 41 0] [2 0 0 0 0 2 1]]
Exp5	10	0.05	50	true	0.88	[[3 0 0 0 1 0 0] [0 20 0 0 0 0 0] [0 0 4 0 4 0 0] [0 0 0 13 0 0 0] [0 0 2 0 7 1 0] [0 0 0 0 0 41 0] [2 1 0 0 1 0 1]]
Exp6	10	0.08	30	True	0.87	[[2 0 0 0 2 0 0] [0 20 0 0 0 0 0] [0 0 4 0 4 0 0] [0 0 0 13 0 0 0] [1 1 0 0 6 0 2] [0 0 0 0 0 41 0] [1 0 0 0 1 1 2]]
Exp 7	10	0.06	100	False	0.93	[[4 0 0 0 0 0 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 1 3 0 6 0 0] [0 0 0 0 0 41 0] [0 0 0 0 0 3 2]]
Exp 8	10	0.06	100	True	0.87	[[4 0 0 0 0 0 0] [1 19 0 0 0 0 0] [0 0 4 0 4 0 0]

						[0 0 0 13 0 0 0] [1 1 1 0 6 0 1] [0 0 0 0 0 41 0] [1 0 0 1 1 1 1]]
Exp 9	15	0.03	150	False	0.94	[[4 0 0 0 0 0 0] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 1 0 8 0 1] [0 0 0 0 0 41 0] [1 0 0 0 0 2 2]]
Exp 10	**15**	**0.1**	**50**	**True**	**0.95**	**[[4 0 0 0 0 0 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 1 0 0 8 1 0] [0 0 0 0 0 41 0] [0 0 0 1 0 2 2]]**
Exp 11	30	0.08	100	True	0.91	[[4 0 0 0 0 0 0] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [1 0 0 12 0 0 0] [1 2 0 0 7 0 0] [0 0 0 0 0 41 0] [3 0 0 0 0 1 1]]
Exp 12	100	0.03	50	True	0.91	[[1 0 0 0 1 2 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 2 1 0 7 0 0] [0 0 0 0 0 41 0] [0 0 0 1 2 0 2]]





Modelul Naive Bayes

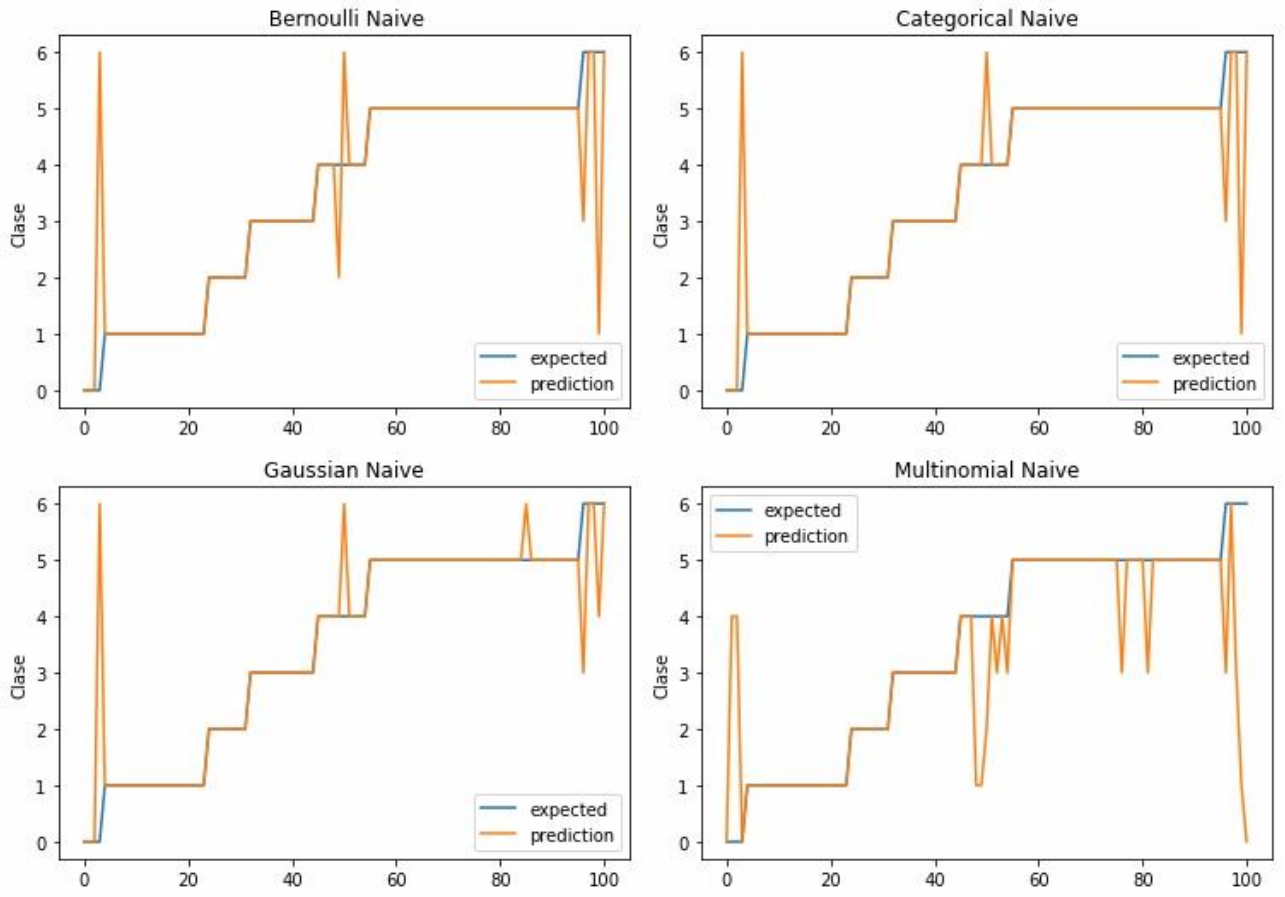
Modelul Naive Bayes este o metodă de învăţare automată bazată pe probabilităţi, care funcţionează prin calcularea probabilităţii asociate cu fiecare clasă pentru o instanţă dată. Apoi, folosind aceste probabilităţi calculate, modelul clasifică instanţa într-o anumită clasă. Numele "Naive" provine din faptul că acest model presupune independenţa între caracteristicile (datele) utilizate în clasificare, ceea ce este o simplificare puternică a realităţii.

- Gaussian Naive Bayes - potrivit pentru date continue care sunt distribuite normal (gaussian). Algoritmul presupune că caracteristicile sunt independente între ele, iar distribuţia datelor pentru fiecare clasă este gaussiană
- Multinomial Naive Bayes- folosit pentru caracteristici care reprezintă frecvenţa cu care anumite evenimente apar într-un set de date
- Bernoulli Naive Bayes- similar cu Multinomial Naive Bayes, dar este folosit pentru variabile binare (0 sau 1)
- Categorical Naive Bayes- similar cu Multinomial Naive Bayes, dar este potrivit pentru caracteristici cu un număr fix de categorii discrete. Acesta consideră frecvenţa cu care apar anumite categorii în setul de date

Experimentele modelului Naive Bayes:

Nume experiment	**Valoarea parametrului k**	**Acuratete medie**	**Matricea de confuzie**
Gaussian	10	0.95	[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 0 0 9 0 1] [0 0 0 0 0 40 1] [0 0 0 1 1 0 3]]
Multinomial	10	0.87	[[2 0 0 0 2 0 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 2 1 2 5 0 0] [0 0 0 2 0 39 0] [1 1 0 2 0 0 1]]
Bernoulli	10	0.95	[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 1 0 8 0 1] [0 0 0 0 0 41 0] [0 1 0 1 0 0 3]]
Categorical	**10**	**0.96**	**[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0]**

			[0 0 0 0 9 0 1] **[0 0 0 0 0 41 0]** **[0 1 0 1 0 0 3]]**



Modelul Knearest Neighbors

Principiul de bază al KNN constă în compararea instanței curente cu ceilalti vecini din setul de date în funcție de o măsură de distanță specificată (euclidean, minkowski,manhattan). Vecinii sunt determinați pe baza valorilor lor de caracteristici și sunt considerați în funcție de clasele din care fac parte.

Procesul de clasificare în KNN constă în următorii pași:

- Se calculează distanța între instanța curentă și fiecare instanță din setul de date.
- Se identifică cei mai apropiați K vecini (K fiind un parametru specificat).
- Se examinează clasele acestor vecini și se determină clasa predominantă sau cea mai comună.
- Instanța curentă este clasificată în acea clasă predominantă.

Parametrii folositi sunt:

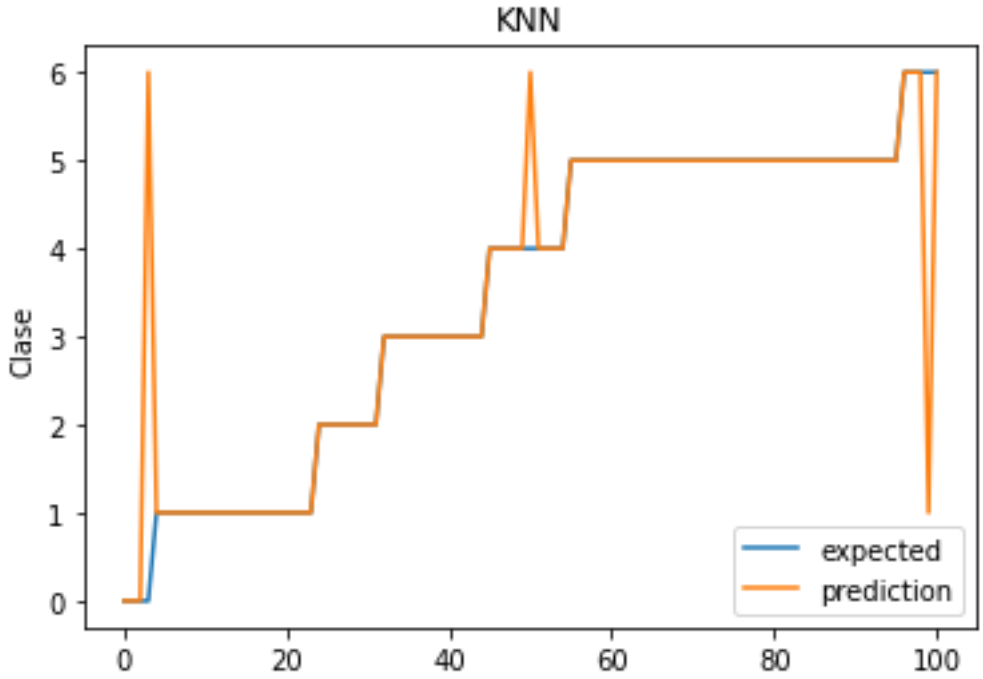
- Parametrul k- numarul de fold-uri

- N_neighbors - numărul de vecini apropiați utilizați pentru a determina clasa unei instanțe noi
- P - controlează puterea metricii utilizate
- Metric- metrica specificată pentru a calcula distanța între puncte
- Algorithm- algoritmul folosit pentru a calcula vecinii apropiați în KNN (ball_tree, kd_tree, brute,auto)

Experimentele modelului Knearest Neighbors:

Nume experiment	Parametrul k	Numar de vecini (n_neighbors)	Parametrul p	Formula aleasa (metric)	Valoare parametru algorithm	Acuratete	Matricea de confuzie
KN 1	5	5	2	minkowski	Ball_tree	0.91	[[4 0 0 0 0 0 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 3 0 7 0 0] [0 0 0 1 0 40 0] [3 1 0 1 0 0 0]]
KN 2	5	10	1	minkowski	Kd_tree	0.85	[[1 0 0 1 1 0 1] [0 20 0 0 0 0 0] [0 0 5 0 3 0 0] [0 0 0 13 0 0 0] [0 0 3 0 7 0 0] [0 0 0 1 0 40 0] [0 2 0 3 0 0 0]]
KN 3	5	10	2	manhattan	Brute	0.85	[[1 0 0 1 1 0 1] [0 20 0 0 0 0 0] [0 0 5 0 3 0 0] [0 0 0 13 0 0 0] [0 0 3 0 7 0 0] [0 0 0 1 0 40 0] [0 2 0 3 0 0 0]]
KN 4	5	10	2	chebyshev	auto	0.67	[[0 0 0 1 0 3 0] [0 17 0 0 0 3 0] [0 0 3 2 3 0 0] [0 0 0 11 0 2 0] [0 0 3 0 0 7 0] [0 0 0 4 0 37 0] [0 0 0 0 0 5 0]]
KN 5	5	10	1	euclidean	auto	0.83	[[0 0 0 1 0 3 0] [0 17 0 0 0 3 0] [0 0 3 2 3 0 0] [0 0 0 11 0 2 0] [0 0 3 0 0 7 0] [0 0 0 4 0 37 0] [0 0 0 0 0 5 0]]
KN 6	5	8	2	minkowski	Auto	0.86	[[1 0 0 1 1 0 1] [0 20 0 0 0 0 0]

							[0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 3 0 7 0 0] [0 0 0 2 0 39 0] [1 2 0 1 0 1 0]]
KN 7	**10**	**1**	**1**	**manhattan**	**Auto**	**0.97**	**[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 0 0 9 0 1] [0 0 0 0 0 41 0] [0 1 0 0 0 0 4]]**



Modelul Support Vector Machine

Acest model SVM se bazează pe identificarea unor vectori de suport în setul de date, care sunt apoi folosiţi pentru a genera hiperplane de separare între clase, cu margini cât mai mari posibil pentru a maximiza precizia clasificării.

Parametrii utilizati pentru SVM sunt:

- Parametrul k- numarul de folduri
- Kernel- funcţie matematică utilizată pentru a transforma datele într-un spaţiu de dimensiuni superioare
- Gamma- responsabilă de transformarea datelor de intrare într-un spaţiu dimensional superior
- C-controlează cât de mult este permisă violarea marginii de separare
- Probability- controlează dacă modelul SVM trebuie să calculeze probabilităţi pentru predicţiile sale

Experimentele modelului Support Vector Machine:

Nume experiment	Parametrul k	Parametrului kernel	Parametrul gamma	Parametrul C	Parametrul probability	Acuratete	Matricea de confuzie
SVM 1	5	linear	scale	0.1	True	0.93	[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 3 0 7 0 0] [0 0 0 0 0 41 0] [0 1 0 1 0 0 3]]
SVM 2	5	poly	Scale	0.5	True	0.88	[[0 0 0 0 0 4 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 0 0 7 3 0] [0 0 0 0 0 41 0] [0 0 0 0 0 5 0]]
SVM 3	5	rbf	Scale	0.1	True	0.62	[[0 0 0 0 0 4 0] [0 18 0 0 0 2 0] [0 0 0 0 0 8 0] [0 0 0 4 0 9 0] [0 0 0 0 0 10 0] [0 0 0 0 0 41 0] [0 0 0 0 0 5 0]]
SVM 4	5	sigmoid	Scale	0.5	False	0.73	[[0 0 0 2 0 2 0] [0 20 0 0 0 0 0] [0 4 0 0 4 0 0] [0 0 0 13 0 0 0] [0 6 1 2 1 0 0] [0 0 0 1 0 40 0] [0 2 0 2 0 1 0]]
SVM 5	5	poly	auto	1.0	True	0.88	[[0 0 0 0 0 3 1] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 0 0 7 3 0] [0 0 0 0 0 41 0] [0 0 0 0 0 5 0]]
SVM 6	**10**	**Linear**	**Auto**	**0.1**	**True**	**0.95**	**[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 2 0 8 0 0] [0 0 0 0 0 41 0] [0 0 0 1 0 0 4]]**
SVM 7	10	Poly	Auto	0.8	False	0.91	[[0 0 0 0 0 3 1] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0]

							[0 0 0 13 0 0 0] [0 0 0 0 9 1 0] [0 0 0 0 0 41 0] [0 0 0 0 0 4 1]]
SVM 8	10	Rbf	Auto	0.9	False	0.93	[[1 0 0 0 0 0 3] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 12 0 0 1] [0 0 1 0 9 0 0] [0 0 0 0 0 41 0] [0 1 0 1 0 0 3]]
SVM 9	10	sigmoid	auto	1	False	0.91	[[1 0 0 0 0 0 3] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 3 0 7 0 0] [0 0 0 0 0 41 0] [0 1 0 1 0 0 3]]
SVM 10	50	rbf	auto	0.4	True	0.90	[[0 0 0 0 1 3 0] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 1 0 9 0 0] [0 0 0 0 0 41 0] [0 1 0 0 0 4 0]]

Modelul arbori de decizie

Arborii de decizie sunt modele de învățare automată care clasifică sau fac predicții pe baza unor reguli de decizie reprezentate sub formă de structuri de tip arbore. Aceste modele stabilesc relații între caracteristicile din setul de date prin efectuarea unor comparații și decizii succesive pe baza valorilor acestor caracteristici.

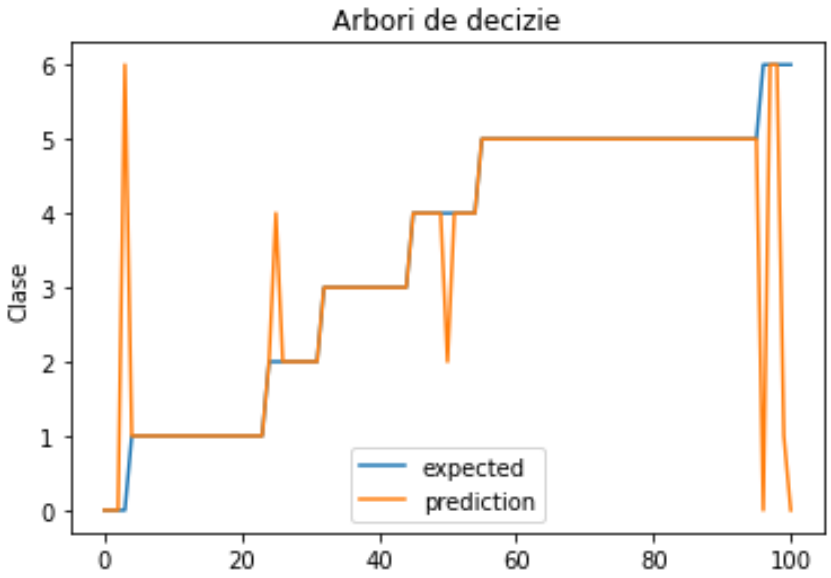
Parametrii folositi ai arborilor de decize sunt:

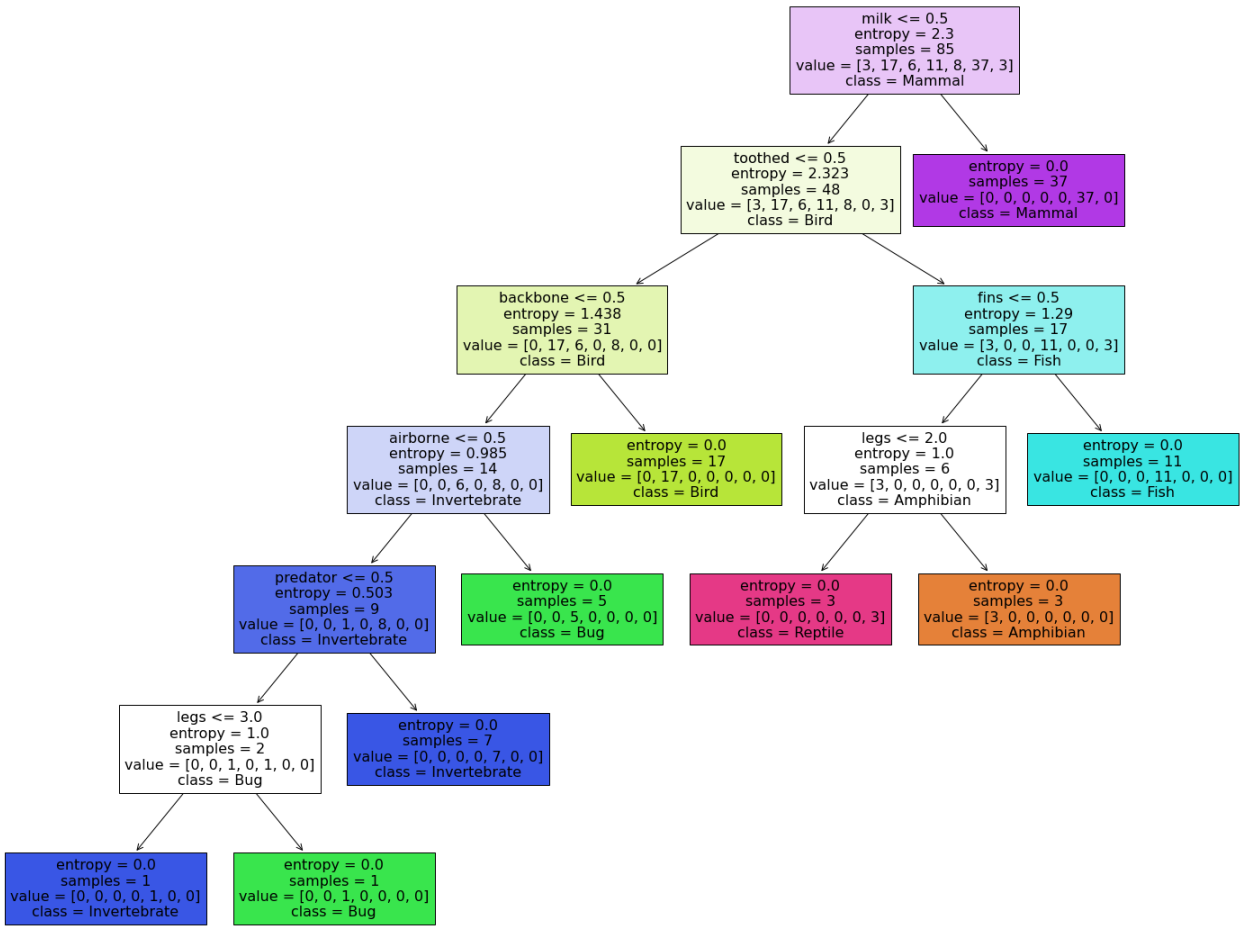
- Parametrul k- nr de folduri
- Max_depth- adâncimea maximă a arborelui de decizie. Arborele nu va fi extins dincolo de această adâncime maximă.
- Min_samples_leaf- numărul minim de eșantioane necesare pentru a fi considerate o frunză (nod terminal) a arborelui. Acesta controlează creșterea arborelui prin stabilirea condițiilor minime pentru divizarea nodurilor. Poate lua valori de tip float din intervalul (0.0, 1.0) si int in intervalul [1,inf].

Experimentele modelului Arbori de decizie:

Nume experiment	**Parametrul k**	**Parametrul criterion**	**Parametrul max_depth**	**Parametrul min_samples_leaf**	**Acuratetea**	**Matricea de confuzie**
Arbore1	5	entropy	None	1	0.92	[[2 0 0 0 0 0 2] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0]

						[0 0 0 13 0 0 0] [0 0 2 0 8 0 0] [0 0 0 0 0 41 0] [3 0 0 0 0 0 2]]
Arbore2	5	Entropy	5	0.2	0.73	[[0 1 0 3 0 0 0] [0 20 0 0 0 0 0] [0 8 0 0 0 0 0] [0 0 0 13 0 0 0] [0 4 3 3 0 0 0] [0 0 0 0 0 41 0] [0 2 0 3 0 0 0]]
Arbore3	5	gini	None	5	0.85	[[0 0 1 0 0 0 3] [0 20 0 0 0 0 0] [0 0 8 0 0 0 0] [0 0 0 13 0 0 0] [0 0 4 0 5 1 0] [1 0 0 0 0 40 0] [3 0 1 0 0 1 0]]
Arbore4	5	gini	10	0.1	0.78	[[0 0 0 0 4 0 0] [0 20 0 0 0 0 0] [0 0 4 0 4 0 0] [0 0 0 13 0 0 0] [0 0 7 0 2 1 0] [0 0 0 0 1 40 0] [0 0 2 0 2 1 0]]
Arbore5	10	Gini	None	3	0.92	[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 5 0 3 0 0] [0 0 0 13 0 0 0] [0 0 2 0 8 0 0] [0 0 0 0 0 41 0] [2 0 0 0 0 0 3]]
Arbore6	**10**	**entropy**	**None**	**1**	**0.94**	**[[3 0 0 0 0 0 1] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 0 0 10 0 0] [0 0 0 0 1 40 0] [1 1 0 1 0 0 2]]**
Arbore7	6	Log_loss	10	0.7	0.41	[[0 0 0 0 0 4 0] [0 0 0 0 0 20 0] [0 0 0 0 0 0 8 0] [0 0 0 0 0 13 0] [0 0 0 0 0 10 0] [0 0 0 0 0 41 0] [0 0 0 0 0 5 0]]
Arbore8	6	gini	10	1	0.86	[[0 0 1 0 0 0 3] [0 20 0 0 0 0 0] [0 0 7 0 1 0 0] [0 0 0 13 0 0 0] [0 0 5 0 5 0 0] [0 0 0 0 0 41 0] [2 0 2 0 0 0 1]]





Daca milk<=0.5 atunci

Daca toothead<=0.5 atunci

Daca Backbone<=0.5 atunci

Daca airbone<=5 atunci

Daca predator<=0 atunci

Daca legs<=3 atunci class4(Invertebrate)

Altfel class 2(bug)

Altfel class 3(Inverebrate)

Altfel class 2(Bug)

Altfel class 1(Bird)

Altfel daca fins<=0.5 atunci

Daca legs<=2 atunci

Class 6(reptile)

Altfel class 0(Amphibian)

Altfel class 4(Fish)

Altfel class5 (mammal)

Concluzii

Din punct de vedere al performanţei, modelul "KN7" (K-Nearest Neighbors) este cel mai eficient în acest context, urmat de modelul "Categorical" (Categorical Naive Bayes), "Exp10" (Perceptron), "SVM6" (Support Vector Machine) şi "Arbore6" (arbore de decizie).

Model	Acuratete
Exp10	0.95
Categorical	0.96
KN7	**0.97**
SVM6	0.95
Arbore6	0.94

Experimentul "Arbore7" în cadrul modelelor de arbori de decizie a obţinut cea mai scăzută performanţă, având o acurateţe de doar 0.41.