

# Tema 2 IA - Modele de prezicere Revenue

Paunoiu Darius Alexandru - 342C4

14 ianuarie 2024

## Cuprins

<b>1</b>	<b>Introducere</b>	<b>2</b>
<b>2</b>	<b>Analiza Date</b>	<b>2</b>
2.1	Analiza Clasa . . . . .	2
2.2	Analiza pe impartirea setului de date . . . . .	3
<b>3</b>	<b>Analiza atribute</b>	<b>4</b>
3.1	Vizualizare atribute numerice . . . . .	4
3.2	Vizualizare atribute categorice . . . . .	5
3.3	Analiza corelare atribute . . . . .	6
3.4	Matrice corelare . . . . .	8
<b>4</b>	<b>Regresia Logistica</b>	<b>9</b>
4.1	Implementarea manuala . . . . .	9
4.2	Comparatie implementare manuala si cea din sklearn . . . . .	9
<b>5</b>	<b>Arbori de decizie</b>	<b>11</b>
5.1	Implementarea manuala . . . . .	11
5.2	Comparatie implementare manuala si cea din sklearn . . . . .	11
<b>6</b>	<b>Analiza tuturor rezultatelor</b>	<b>13</b>

# 1 Introducere

Tema isi propune implementarea a 2 tipuri de algoritmi de invatare automata, regresia logistica si arborii de decizie, atat manual, cat si de mana, pentru a prezice daca o anumita persoana va face sau nu o achizitie.

Toate testelete au fost rulate pe acelasi set de 10 impartiri aleatorii. Sistemul pe care au fost rulate este:

- Procesor: Ryzen 7 1700 3.0 GHz 3.5GHz Boost
- Memorie: 16 GB Ram DDR4 2800MHz

Toate datele au fost salvate folosind pickle, astfel nu este necesara rerularea algoritmilor pentru cazurile de teste, acestea importandu-se automat daca sunt gasite fisierele. Tema este impartita in 3 notebook-uri: data.ipynb, regression.ipynb, trees.ipynb si comparison.ipynb

## 2 Analiza Date

### 2.1 Analiza Clasa

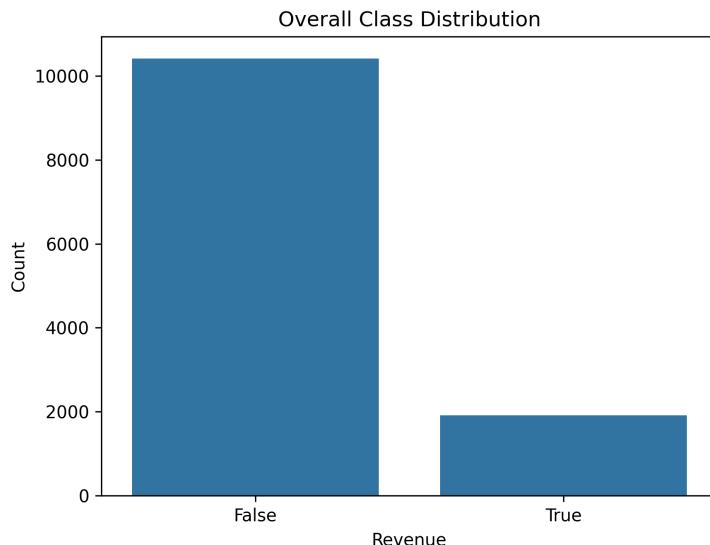


Figura 1: Distributie clasa.

In primul rand putem vedea ca clasa in sine nu este echilibrata, avand mult mai multe valori false decat adevarate, dar acest lucru este firesc considerand mediul din care setul provine, majoritatea lumii necumparand un produs.

## 2.2 Analiza pe impartirea setului de date

Vom analiza media impartirii setului de date pe test/antrenare.

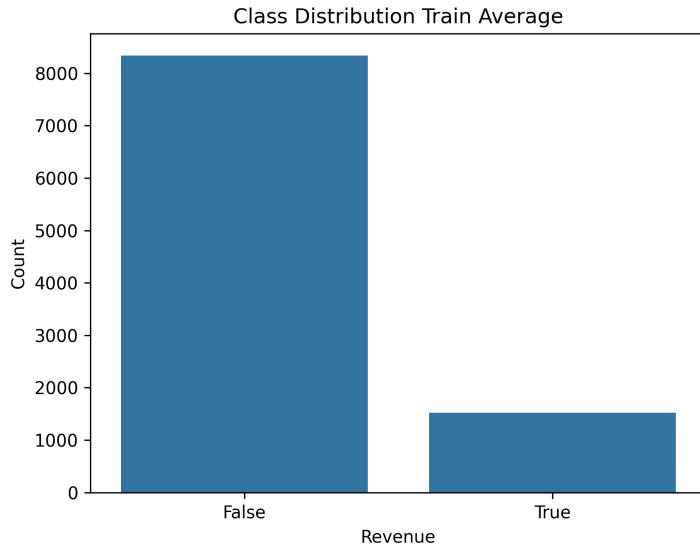


Figura 2: Distributie clasa set antrenare.

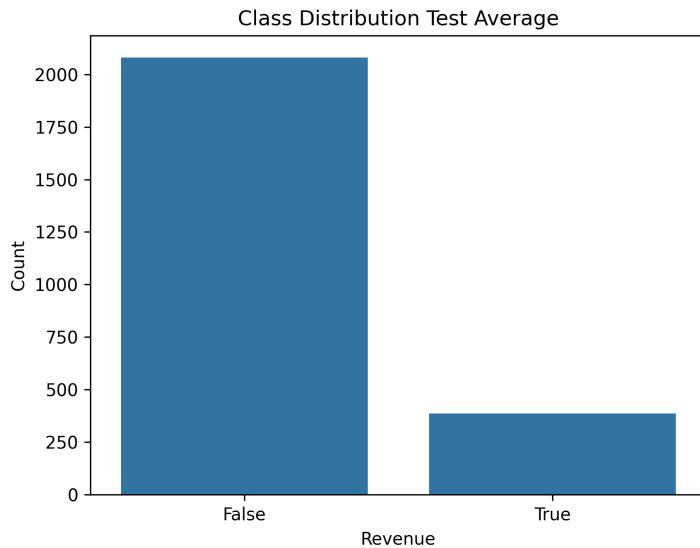


Figura 3: Distributie clasa set teste.

Se poate observa cum proportional vorbind, impartirii este aceeasi. Asta inseamna ca nu vom avea cazuri speciale in antrenare si testare, precum cel in care pe antrenare avem doar valori false, iar pe test doar valori de adevar, ceea ce este un lucru bun.

### 3 Analiza atributelor

#### 3.1 Vizualizare atributelor numerice

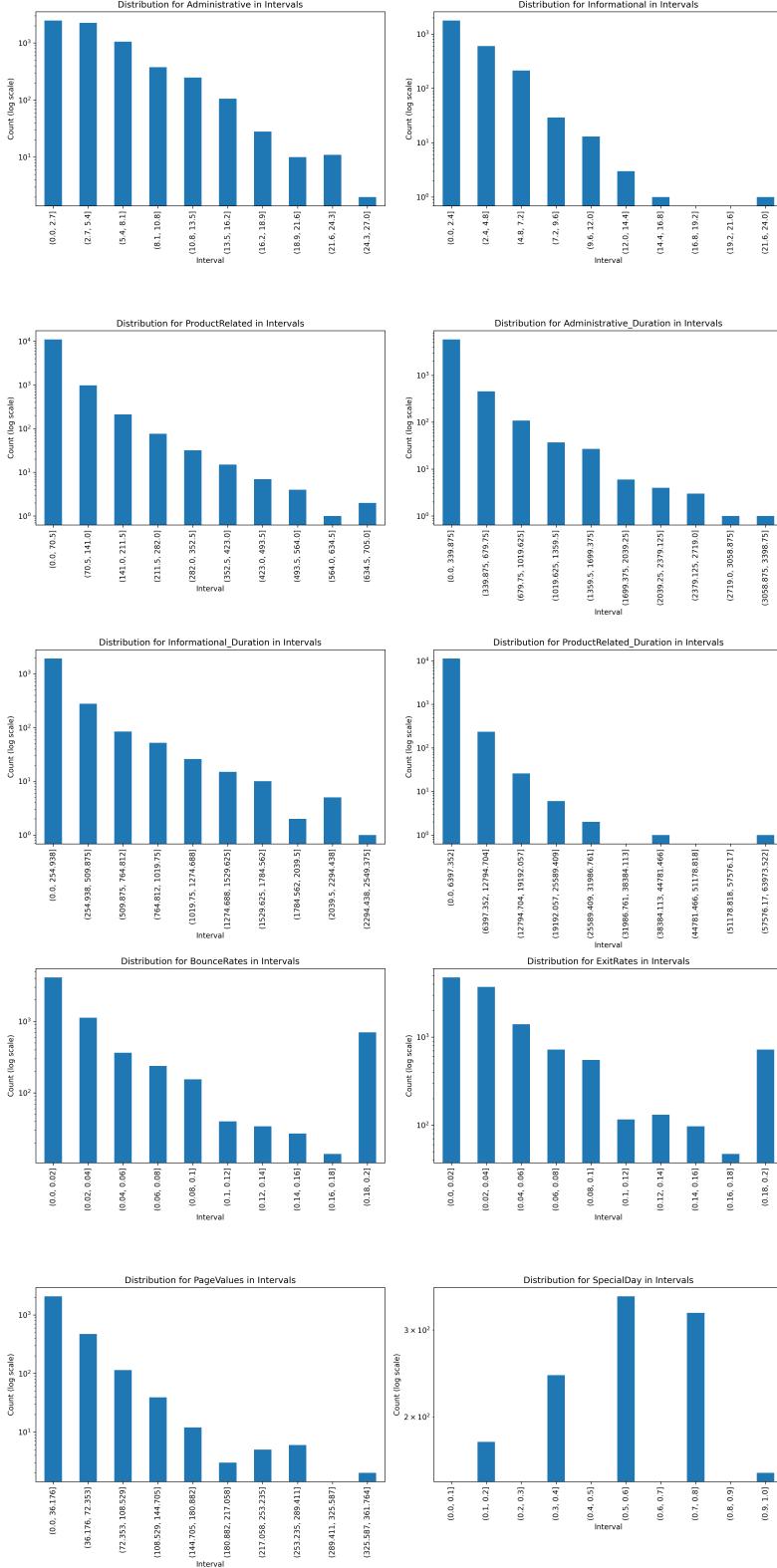


Figura 4: Distributie atributelor numerice.

Impartirea a fost facuta pe intervale de granularitate 10%. Din aceste grafice se poate observa cum niciun atribut numeric nu pare echilibrat, lucru care din nou este natural. De exemplu, majoritatea tinde sa stea cat mai putin pe o pagina administrativa. Desi nu este chiar echilibrat, pot spune ca nu exista o anomalie in acest set de date.

### 3.2 Vizualizare atribute categorice

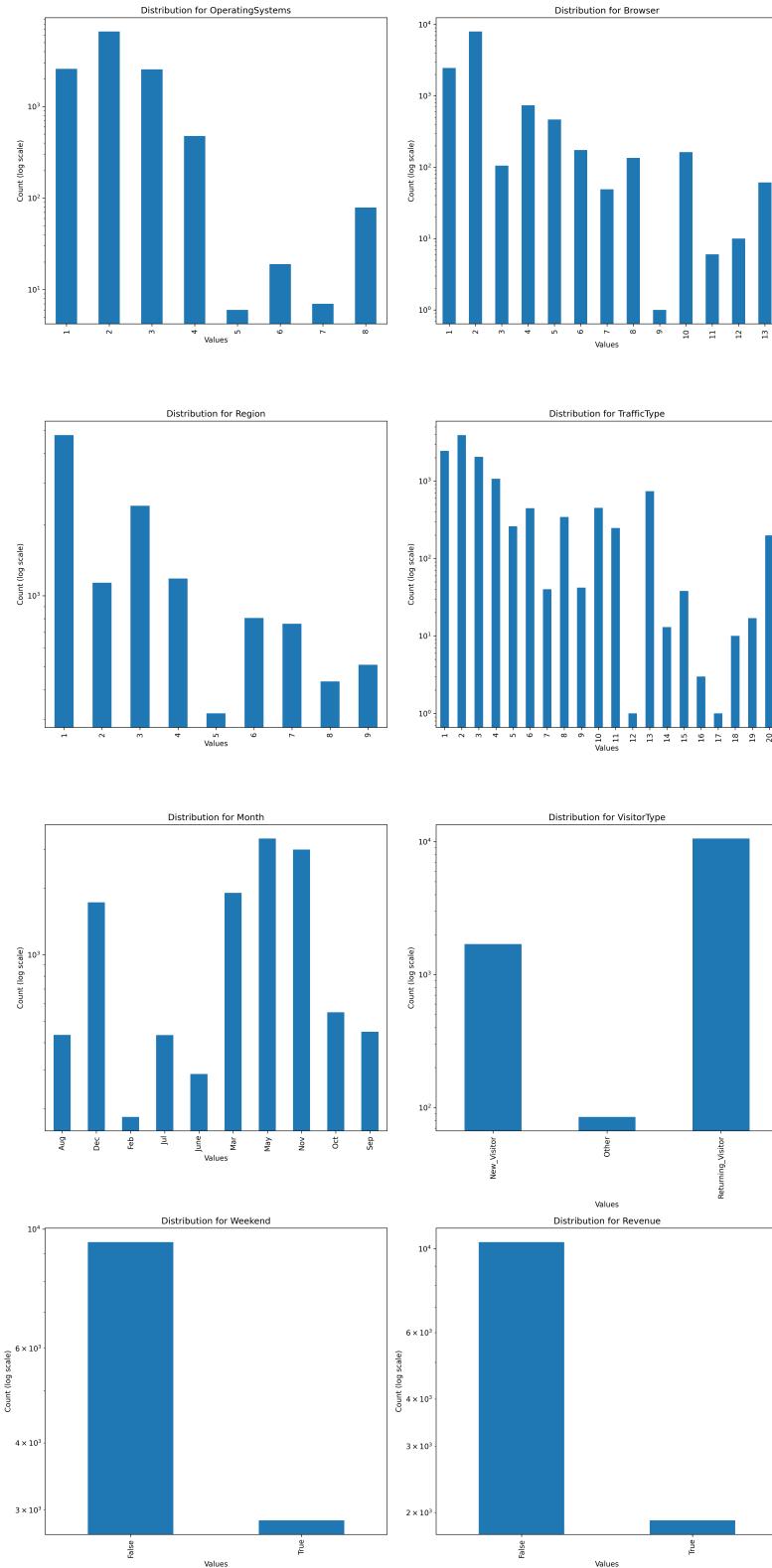


Figura 5: Distributie atribute categorice.

Similar ca la cele numerice, nu pot spune ca vad vreo anomalie in aceste grafice. Nu sunt echilibrate evident, dar nu ar trebui sa fie. Pot spune ca setul de date este unul curat.

### 3.3 Analiza corelare atribute

Valorile pentru Point Biserial Correlation sunt:

Attribute	Correlation	P-Value
Administrative	0.138917	3.519760e-54
Informational	0.095200	3.174034e-26
ProductRelated	0.158538	3.241187e-70
Administrative_Duration	0.093587	2.146514e-25
Informational_Duration	0.070345	5.282871e-15
ProductRelated_Duration	0.152373	6.115338e-65
BounceRates	-0.150673	1.594198e-63
ExitRates	-0.207071	1.662654e-119
PageValues	0.492569	0.000000e+00
SpecialDay	-0.082305	5.498934e-20

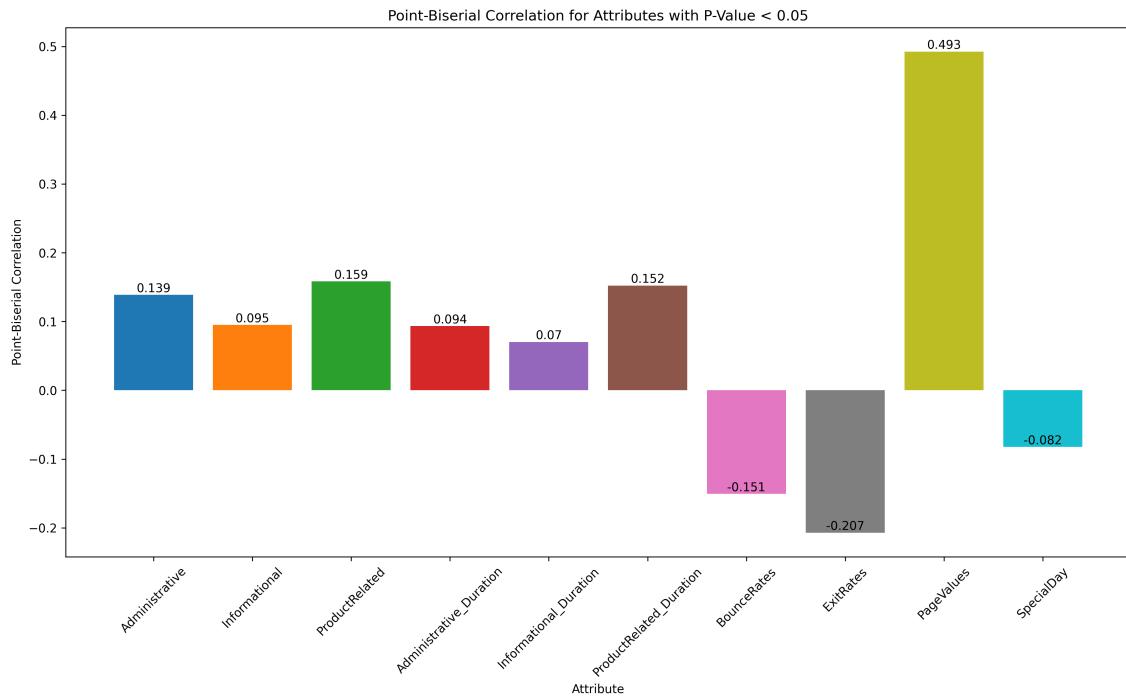


Figura 6: Corelare Point Biserial.

Se poate observa cum pentru majoritatea atributelor numerice, gradul de corelare este foarte mic. Totusi, atribute precum Administrative si ProductRelated par sa fie mai relevante. Lipsa corelarii a multor atribute va putea fi observata la arbori, deoarece rezultatele ar trebui sa nu fie atat de afectate de adancimea arborilor.

Valorile pentru ChiSquared Correlation sunt:

Attribute	Correlation	P-Value
Weekend	10.390978	1.266325e-03
OperatingSystems	75.027056	1.416094e-13
Browser	27.715299	6.087543e-03
Region	9.252751	3.214250e-01
TrafficType	373.145565	1.652735e-67
Month	384.934762	2.238786e-77
VisitorType	135.251923	4.269904e-30

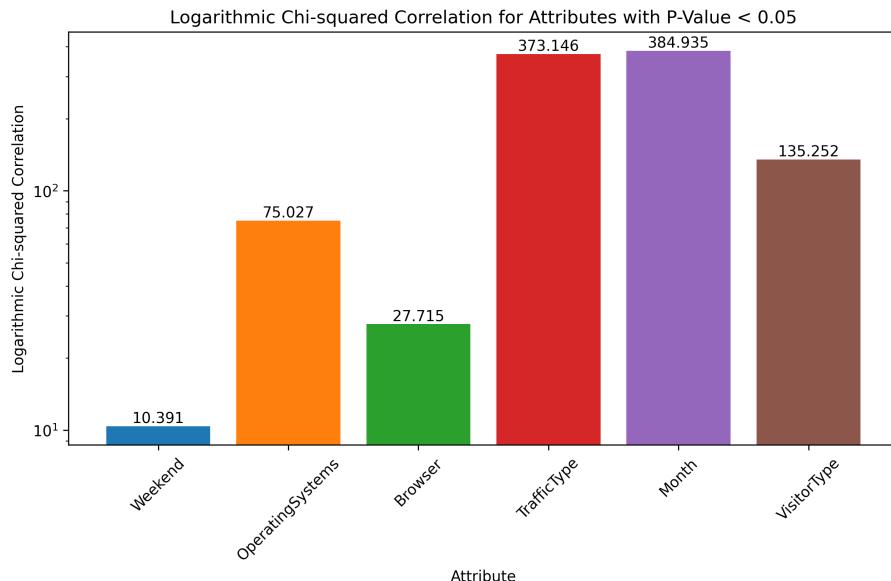


Figura 7: Corelare Chi Squared.

Aici vedem putin mai multa influenta comparativ cu cele numerice, in special daca ne uitam la TrafficType sau Month. Am putea spune ca atributele categorice sunt mai relevante, dar este normal ca un atribut precum Month sa aiba o influenta mare, deoarece in perioada sarbatorilor lumea face mai multe achizitii.

### 3.4 Matrice corelare

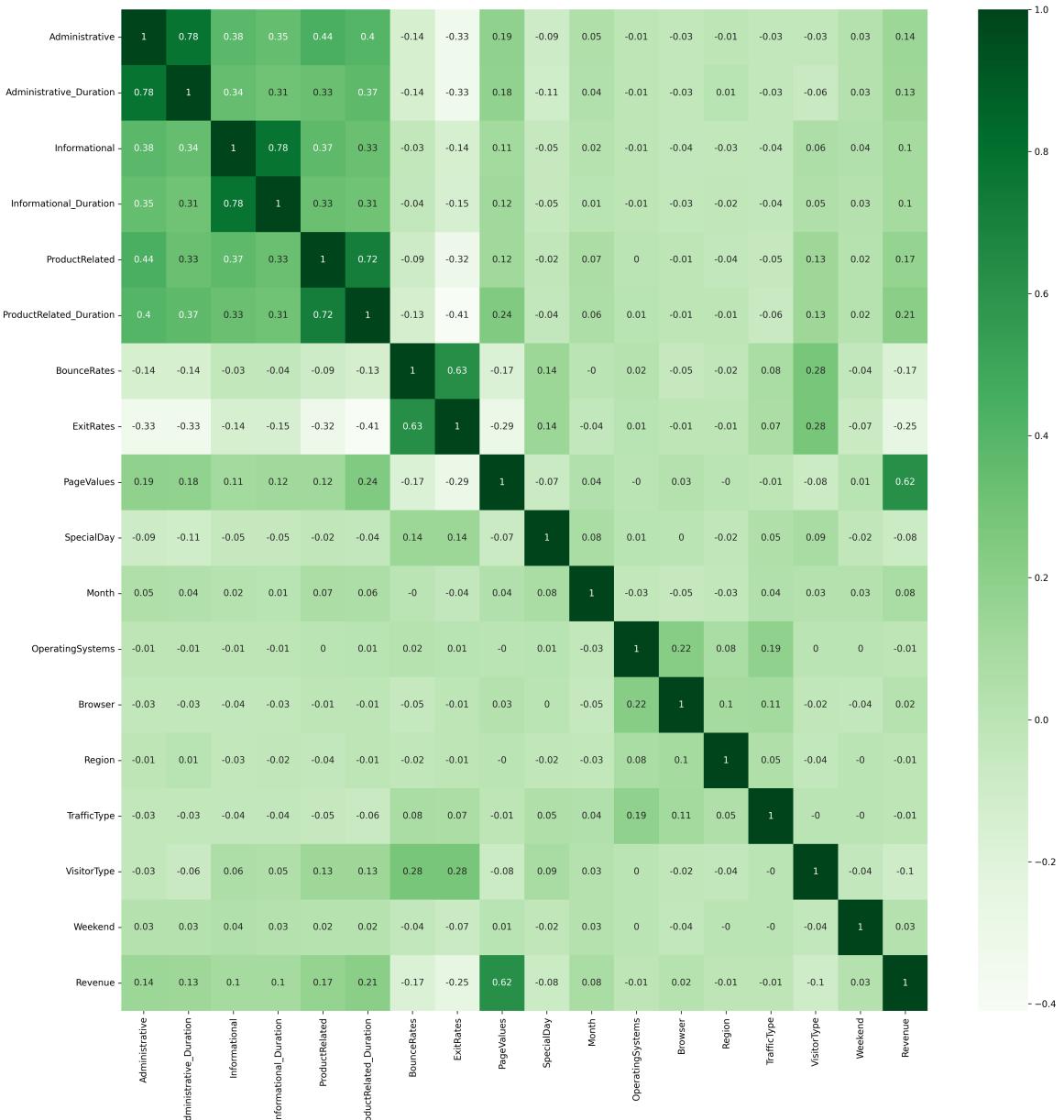


Figura 8: Matrice corelare.

Am decis sa contruiesc si o matrice de corelare, care nu este totusi bazata pe numerele de mai sus, ci este facuta automat de seaborn. Putem totusi observa acelasi trend, cum ca multe atribute au o influenta mica, aproape spre zero. In acest caz, cea mai mare influenta o are PageValues.

## 4 Regresia Logistica

### 4.1 Implementarea manuala

Implementarea pleaca de cea folosita la laborator. De fapt, este aproape identica, dar s-a folosit excep din scipy pentru functia logistica pentru o precizie mai buna.

### 4.2 Comparatie implementare manuala si cea din sklearn

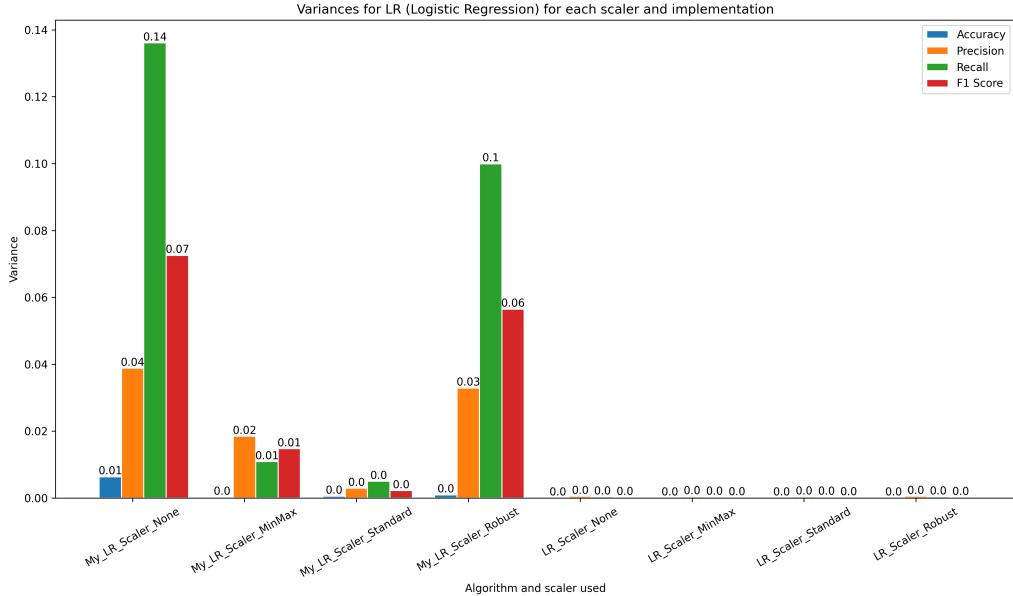


Figura 9: Comparatie varianta metrii intre cele 2 implementari ale regresiei logistice.

In primul rand, putem observa ca dupa cele 10 rulari, varianta metricilor este foarte mica. Cea mai mare variatie este 0.14, pentru recall cand nu folosim niciun scaler pentru implementarea manuala. Asta inseamna ca impartile pe care le-am avut au fost la fel de echilibrate.

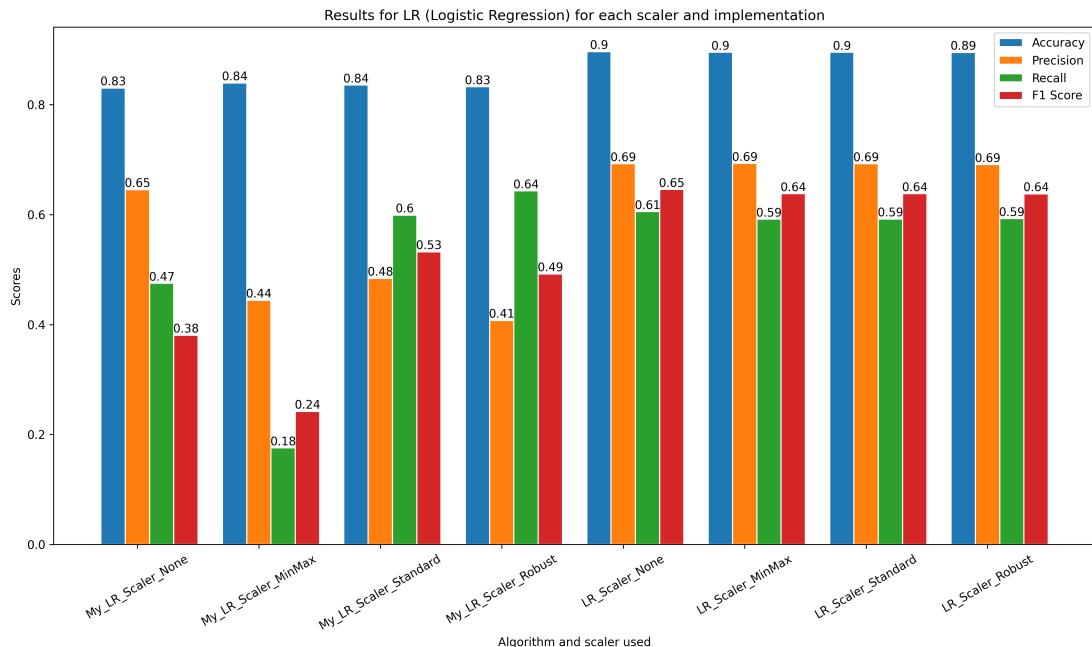


Figura 10: Comparatie medie metrii intre cele 2 implementari ale regresiei logistice.

Se poate observa ca varianta din sklearn are mereu rezultate mult mai bune decat implementarea manuala, lucru ce era de asteptat, deoarece nu exista optimizari in cea manuala. Cu toate acestea,

avem totusi o performanta foarte buna, mai ales cand folosim un scaler precum StandardScaler sau RobustScaler. Motivul pentru care cred eu ca MinMaxScaler are rezultate foarte slabe, este acela ca setul de date nu este chiar echilibrat, iar atunci o scalare directa la maxim si minim va duce la multe extremitati.

## 5 Arbori de decizie

### 5.1 Implementarea manuala

Implementarea pleaca de cea folosita la laborator, dar a deviat foarte mult. Deoarece a trebuit sa impart pe intervale numerice, am decis ca toate atributele sa fie encodeate numeric, pentru a usura implementarea si a lucra doar cu atribute numerice. De asemenea, implementarea are ca punct de plecare atat resursa din enunt pentru impartirea binara pe intervale cat si o alta implementare publica, link-ul fiind mentionat in bibliografie.

### 5.2 Comparatie implementare manuala si cea din sklearn

Vreau sa mentionez ca am realizat grafice si pentru adancimele 4 si 5, dar adaugarea lor nu va aduce informatie noua. Ambele variante ale implementarii nu-si schimba comportamentul pe acele adancimi, deci voi exclude variaitia lor. Mai mult, se poate vedea cum ca variatiile sunt aproape 0, deci algoritmi sunt aproape independenti de impartire.

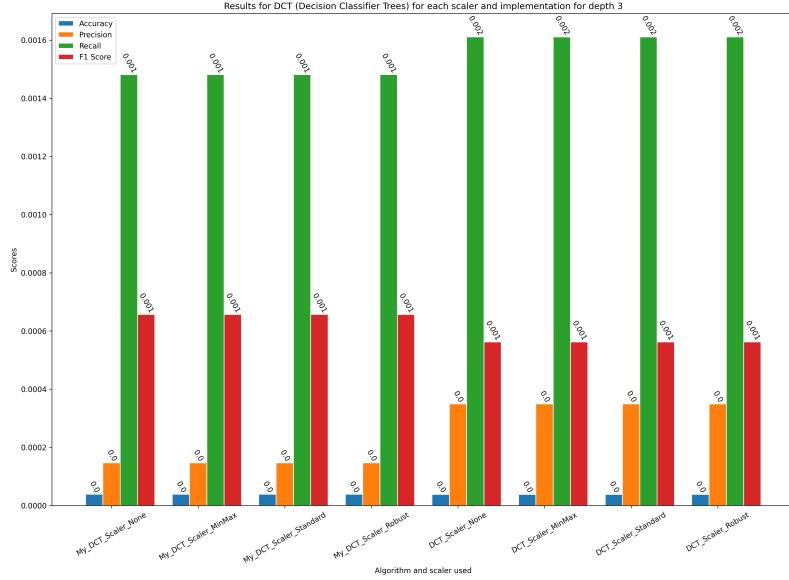


Figura 11: Comparatie variatie metrici intre cele 2 implementari ale arborilor de decizie pentru adancime 3.

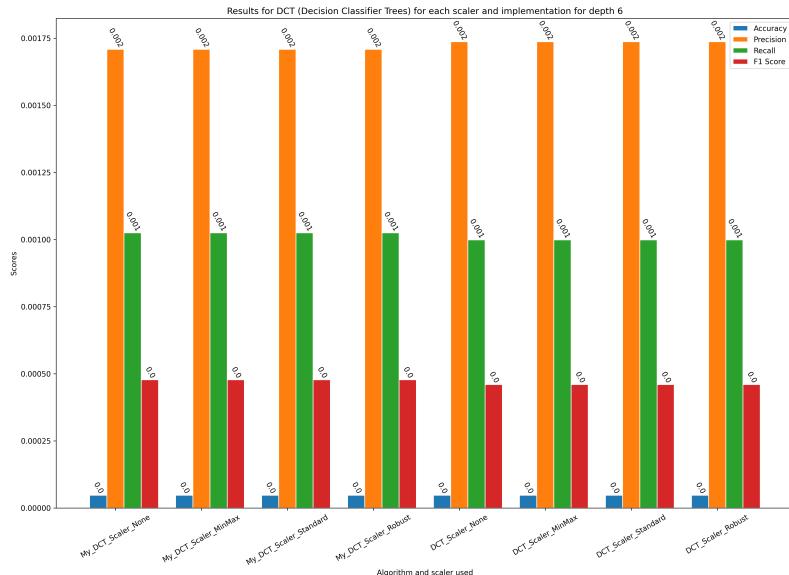


Figura 12: Comparatie variatie metrici intre cele 2 implementari ale arborilor de decizie pentru adancime 6.

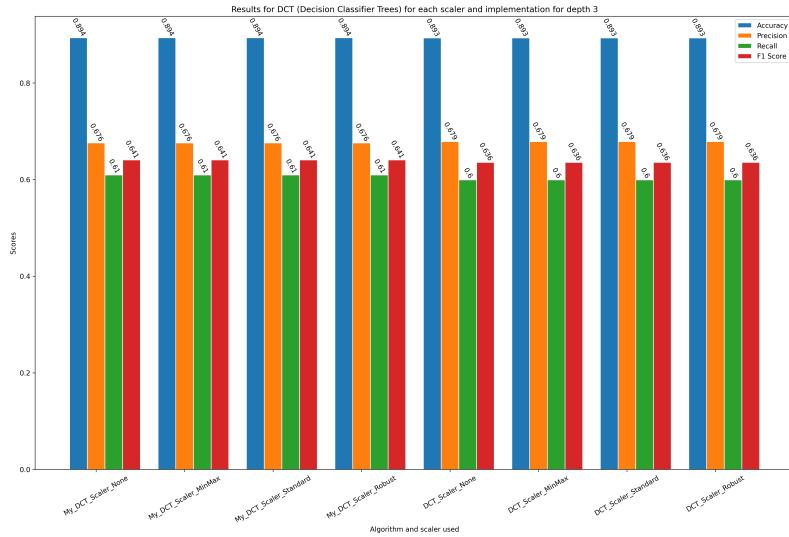


Figura 13: Comparatie mediei metrici intre cele 2 implementari ale arborilor de decizie pentru adancime 3.

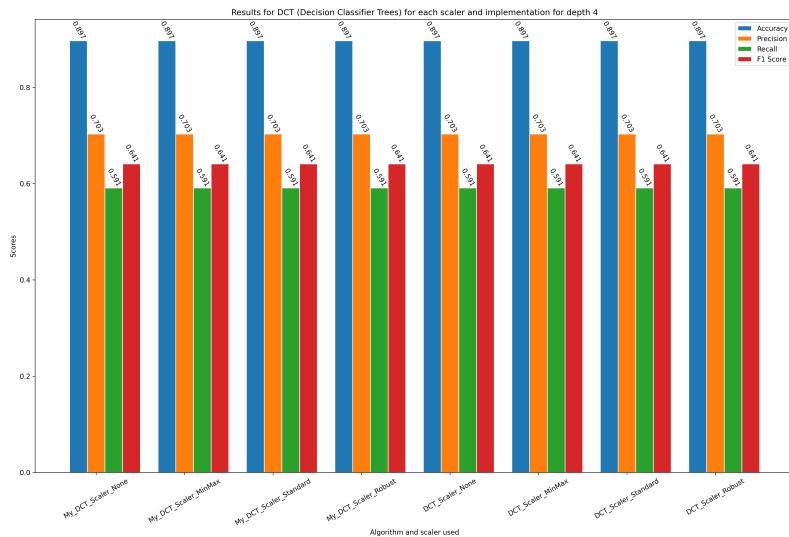


Figura 14: Comparatie mediei metrici intre cele 2 implementari ale arborilor de decizie pentru adancime 3.

In primul rand, dupa cum se poate vedea, adancimea arborelui nu influenteaza aproape deloc arborele, dupa cum am mentionat mai sus. Rezultatele difera foarte putin, diferentele fiind sub 2%, si variind (de exemplu acuratetea creste, dar f1 scade). De asemenea, pe ambele cazuri, se poate observa cum scalarea nu influenteaza absolut deloc rezultatul arborilor. Acest lucru era de asteptat, deoarece imparitile ar trebui sa fie identice indiferent de scalare.

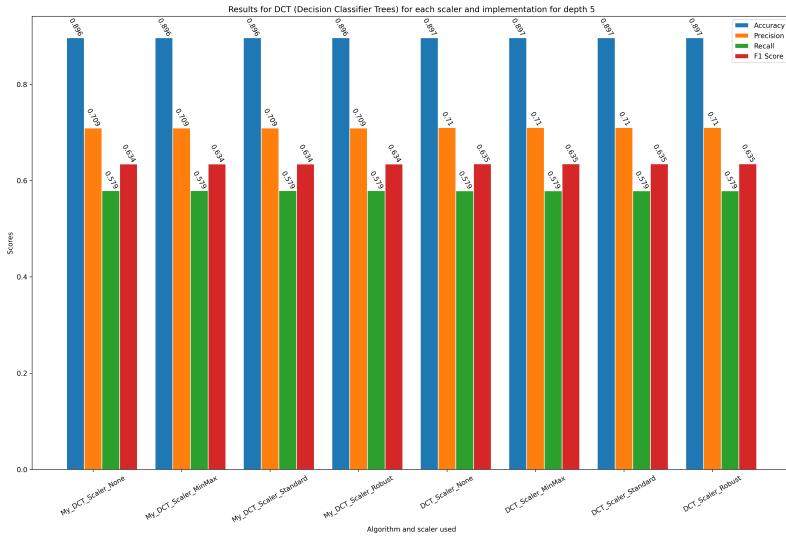


Figura 15: Comparatie mediei metrici intre cele 2 implementari ale arborilor de decizie pentru adancime 3.

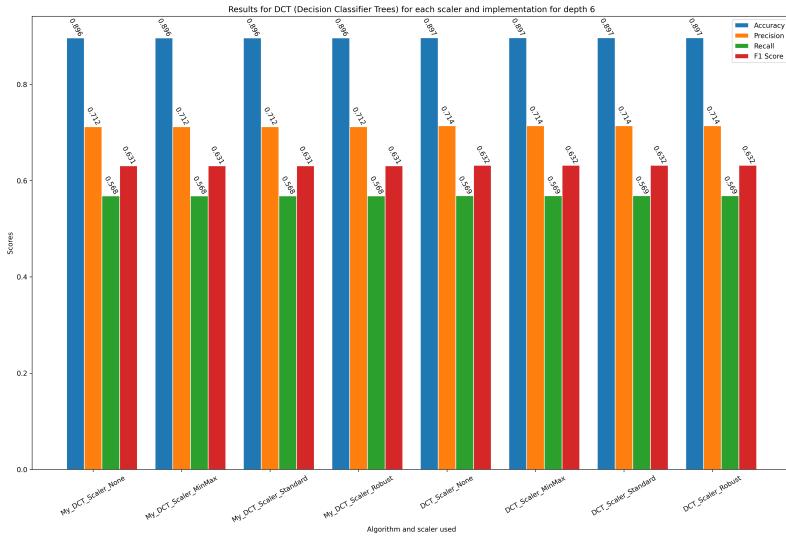


Figura 16: Comparatie mediei metrici intre cele 2 implementari ale arborilor de decizie pentru adancime 6.

## 6 Analiza tuturor rezultatelor

Dupa ilustrarea acestor date se poate observa ca cel mai bun algoritm de regresie logistica este cel din sklearn fara a folosi nicio scalare (diferenta foarte mica de 0.01, dar avem un recall mai bun deci un F1 mai bun) cu un scor de 0.65, iar pe partea de arbori de decizie majoritatea au aceeasi performanta, castigatorii fiind implementarea manuala de arbori cu adancime 3 sau implementarea din sklearn cu adancime 4, ambele avand un scor de 0.641. Rezultatele sunt foarte apropiate. Nu stiu exact ce optimizari are regresia logistica din sklearn astfel incat fara nicio scalare sa obtina cel mai bun scor, dar din nou diferenta este aproape infima. De mentionat este ca arborii au in timp mai mare de antrenare, in special varianta manuala creata de mine, care dureaza foarte mult, nefolosind tehnici de vectorizare. Din punctul meu de vedere regresia logistica este cel mai bun algoritm, cel putin pentru acest caz, deoarece are un timp de antrenare foarte mic iar rezultate foarte bune. Tind sa cred ca arborii de decizie devin din ce in ce mai relevanti pe masura ce numarul de atribute din setul de date creste.

## **Bibliografie**

### **Bibliografie**

- [1] Laboratoarele si cursurile de Inteligenta Artificiala, CTI, ACS, UPB.
- [2] Resursele din cerinta temei.
- [3] Resursa publica pentru implementare arbori de decizie:  
<https://github.com/AyanPahari/Decision-Tree-from-Sratch>