Universitatea „Alexandru Ioan Cuza” Iași

Facultatea de Informatică

LUCRARE DE LICENȚĂ

O analiză asupra algoritmilor de clasificare discriminativi și generativi și aplicarea acestora în diagnoza cancerului la sân

propusă de

Giurgilă Andreea

Sesiunea: Iulie, 2018

Coordonator științific

Conferențiar, Dr. Liviu Ciortuz

Universitatea „Alexandru Ioan Cuza” Iași
Facultatea de Informatică

O analiză asupra algoritmilor de clasificare discriminativi și generativi și aplicarea acestora în diagnoza cancerului la sân

Giurgilă Andreea

Sesiunea: Iulie, 2018

Coordonator științific
Conferențiar, Dr. Liviu Ciortuz

Avizat,

Îndrumător Lucrare de Licenţă

Titlul, Numele şi prenumele_____

Data _____Semnătura_____

DECLARAŢIE privind originalitatea conţinutului lucrării de licenţă/diplomă/disertaţie/absolvire

Subsemnata **GIURGILĂ ANDREEA** domiciliul în **ADJUDENI, NEAMT**

născut(ă) la data de **03 FEB 1997**, identificat prin CNP **2970203270831**, absolvent(a) al(a) Universităţii „Alexandru Ioan Cuza" din Iaşi, Facultatea de Informatică promoţia 2018, declar pe propria răspundere, cunoscând consecinţele falsului în declaraţii în sensul art. 326 din Noul Cod Penal şi dispoziţiile Legii Educaţiei Naţionale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licenţă cu titlul *"O analiză asupra algoritmilor de clasificare discriminativi și generativi și aplicarea acestora în diagnoza cancerului la sân*" elaborată sub îndrumarea dl. Conf. Dr. Liviu Ciortuz, pe care urmează să o susţină în faţa comisiei este originală, îmi aparţine şi îmi asum conţinutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licenţă/diplomă/disertaţie/absolvire să fie verificată prin orice modalitate legală pentru confirmarea originalităţii, consimţind inclusiv la introducerea conţinutului său într-o bază de date în acest scop.

Am luat la cunoştinţă despre faptul că este interzisă comercializarea de lucrări ştiinţifice in vederea facilitării fasificării de către cumpărător a calităţii de autor al unei lucrări de licenţă, de diploma sau de disertaţie şi în acest sens, declar pe proprie răspundere că lucrarea de faţă nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, …… Semnătură student ……

DECLARAŢIE DE CONSIMŢĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licenţă cu titlul „*O analiză asupra algoritmilor discriminativi și generativi și aplicarea acestora în diagnoza cancerului la sân*”, codul sursă al programelor şi celelalte conţinuturi (grafice, multimedia, date de test etc.) care însoţesc această lucrare să fie utilizate în cadrul Facultăţii de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea Alexandru Ioan Cuza Iaşi să utilizeze, modifice, reproducă şi să distribuie în scopuri necomerciale programele calculator, format executabil şi sursă, realizate de mine în cadrul prezentei lucrări de licenţă.

Iaşi, *29.06.2018*

Giurgilă Andreea

_____

(semnătura în original)

MULŢUMIRI

În primul rând, aș dori să mulțumesc coordonatorului meu științific, Dr. Conf. Liviu Ciortuz. Fără ajutorul și implicarea lui nu aș fi reușit să scriu această lucrare. De asemenea aș dori să-i mulțumesc pentru introducerea pe care mi-a oferit-o în acest domeniu în cadrul cursului de Învățare Automată de la Facultatea de Informatica din Iași. Acest curs a fost cel care m-a determinat să aprofundez noțiunile introduse la curs și să le extind într-o lucrare de cercetare.

Un ajutor foarte mare în scrierea acestei lucrări au fost cursurile și materialele domnilor profesori Andrew Ng. (Stanford) și Tom Mitchell (Carnegie Mellon University). Deși aceștia nu au fost profesorii mei, resursele postate online m-au ajutat în tratarea anumitor subiecte.

De asemenea, sunt recunoscătoare Facultății de Informatică din Iasi pentru oferirea resurselor necesare scrierii acestei lucrari.

Cuprins

1 Introducere

Odată cu creşterea exponenţială a cantităţii de date existente în lume, necesitatea unor metode potrivite de analiză a acestor date creşte şi ea. Prin analiză înţelegem învăţarea unui model dintr-un set de date cu ajutorul căruia să putem prezice pentru un set de date noi un anumit comportament. Prin crearea unor modele probabiliste, învăţarea Automată are ca scop automatizarea acestui proces de analiză.

Lucrarea de faţă este structurată pe patru capitole care vor ajuta la întelegerea modului în care algoritmii generativi diferă faţă de cei discriminativi şi cum se pot aplica aceştia în rezolvarea unei probleme din viaţa reală şi anume diagnoza cancerului la sân.

Capitolul I cuprinde o scurtă descriere a modului în care lucrează un algoritm de clasificare generativ. În special, voi prezenta algoritmul Bayes Naiv şi o variaţie a acestuia, Bayes Naiv Gaussian. De asemenea voi arăta cum forma unuia dintre parametrii acestui algoritm se aseamană cu un parametru dintr-un algoritm discriminativ - Regresia Logistică.

Capitolul II prezintă modul de lucru al algoritmilor de clasificare discriminativi. În special, acest capitol prezintă doi algoritmi: Regresia Logistică şi Maşinile cu Vector Suport.

Capitolul III descrie cateva aplicaţii ale algoritmilor specificaţi mai sus, punând în evidentă diferenţa dintre aceştia. Voi aplica un algoritm de clasificare discriminativ şi unul generativ pe acelaşi set de date si voi compara rezultatele.

În **Capitolul IV** voi aplica algoritmii discutaţi în această lucrare în rezolvarea unei probleme din viata reala: diagnoză a cancerului la sân. Voi prezenta modul de abordare al acestei probleme, cum se face selecţia de atribute şi voi interpreta rezultatele obţinute.

2 Modelul Generativ

2.1 Introducere capitol

Un model generativ este un model care, folosindu-se de un model probabilistic, descrie modul în care sunt generate datele.

Presupunem că avem o problemă de clasificare în care ne dorim să deosebim între motociclete şi maşini. Pentru a învaţă pe cineva să recunoască o maşină desenăm pe o foaie de hârtie modelul unei maşini şi pe o foaie separată cel al unei motociclete. Când ne confruntăm cu problemă de a clasifica un nou obiect, trebuie doar să decidem cu care dintre modelele desenate anterior este mai asemănător obiectul. Acest tip de clasificare se încadrează în categoria algoritmilor de învăţare generativi. Mai formal, numim algoritm generativ un algoritm care învaţă o funcţie scor f(x, y) a datelor de intrare x şi etichetele y. Pentru a prezice o nouă instanţă y, algoritmul alege y_i pentru care funcţia f(x, y_i) este maximă.

2.2 Teorema lui Bayes pentru învăţarea de etichete (ipoteze)

Fiind dată o instanţă x_i şi eticheta corespunzătoare x_i, formula lui Bayes ne prezintă o relaţie intre probabilitatea lui x_i atunci când s-a observat deja eticheta y_i (probabilitatea lui x_i a posteriorii observării y_i) si probabilitatea ca rezultatul clasificării sa fie y_i, ţinând cont ca instanta de clasificat este x_i (probabilitatea lui y_i a posteriorii lui x_i). Mai precis:

$$p(x_i \mid y_i) = \frac{p(y_i \mid x_i)p(x_i)}{p(y_i)} \tag{1}$$

Unde $p(x_i)$ este probabilitatea a priorii a lui x_i iar $p(y_i)$ este probabilitatea a priorii a lui $p(x_i)$.

2.2.1 Estimarea probabilităţilor (MLE şi MAP)

Atunci când ne dorim sa estimam un model de distribuţie probabilistic al datelor de antrenare, exista doua metode populare de abordare: estimarea în sensul verosimilităţii maxime (MLE - en: Maximum Likelihood Estimation) şi estimarea în sensul probabilităţii maxime a posteriorii (MAP

- en: Maximum A Posterior).

Pentru a deosebi între cele două moduri de abordare ne vom folosi de un exemplu simplu*: aruncarea unei monezi. Vom nota cu M variabila care ne dă rezultatul aruncării, cu p = p(M=ban), p∈(0,1), probabilitatea ca rezultatul aruncării să fie ban (1-p=p(M=stema) va fi probabilitatea ca rezultatul să fie stema) şi cu p̂ probabilitatea estimată ca rezultatul să fie ban.

Dupa n aruncări, obţinem de a1ori ban si de a_2 ori stema. Intuitiv, deducem ca p = $\frac{a_1}{a_1+a_2}$. Acest mod de gandire este unul bun atunci cand datele de antrenare sunt suficiente. In caz contrar, acest algoritm ne poate conduce spre rezultate eronate. De exemplu, daca aruncand moneda de 4 ori obtinem in mod consecutiv ban, vom deduce ca p(M=ban)=1 ceea ce este putin probabil sa se intample daca stim ca folosim o moneda cinstita care are P(M=ban)∼0.5. Acest argument sta la baza celei de-a doua metoda de abordare.

A doua metoda ne permite să ne folosim de presupuneri sau cunostinte a priori despre moneda pentru a obtine un rezultat cat ai aproape de adevar. In cazul nostru, putem introduce un numar imaginar b1 de rezultate ban respectiv un numar b2 de aruncari in urma carora a fost observata stema. Aceste numere vor fi alese conform presupunerii a priorii pe care o avem despre moneda. De exemplu, presupunând ca moneda este una cinstita şi ca P(M=ban)=0.5 putem alege $b_1 = b_2 = 5$. Astfel noua estimare a lui $\hat{p}$ devine: $\hat{p} = \frac{a_1+b_1}{(a_1+b_1)+(a_2+b_2)}$.

In continuare, vom demonstra ca cele doua metode prezentate mai sus corespund estimării probabilităţii în sens MLE, respectiv MAP.

2.2.1.1 Estimarea MLE Estimarea MLE calculează parametrii unui model care explica cel mai bine datele de antrenare D. Prin definiţie:

$$\hat{p} = argmax_p P(D \mid p) \tag{2}$$

Definim variabila aleataore M astfel:

$$M = \begin{cases} 1, & \text{daca rezultatul dupa o aruncare e ban} \\ 0, & \text{altfel} \end{cases}$$

De asemenea, notam cu θ probabilitatea adevărata ca rezultatul după o aruncare este ban. Observăm că, dacă avem o singura aruncare, p(D $|\theta$) = dacă M=1 respectiv p(D $|\theta$)=1-θ dacă M=0. Presupunem că după n aruncări consecutive şi independente obţinem de b_1 ori rezultatul M = 1 si de b_2 ori rezultatul M = 0. Putem calcula P(D $|\theta$) înmulţind fiecare probabilitate. Vom avea:

$$P(D \mid \theta) = \theta^{b_1}(1-\theta)^{b_2} \tag{3}$$

Folosind estimatorul MLE, ne dorim sa determinam valoarea lui care maximizează P(D $|\theta$). Acest lucru este echivalent cu maximizarea logaritmului:

$$f(\theta) = argmax_\theta P(D \mid \theta) = argmax_\theta lnP(D \mid \theta) \tag{4}$$

Pentru a găsi maximul, calculam derivata funcţiei de mai sus şi o egalam cu 0.

$\frac{d}{d\theta}$ f(θ) = $\frac{d}{d\theta}$ ln P(D $|\theta$)

=$\frac{d}{d\theta}\ln(\theta^{b_1}(1-\theta)^{b_2})$

= $\frac{d}{d\theta}$[ln $\theta^{b_1} + ln(1-\theta)^{b_2}$]

= $\frac{d}{d\theta}$[b$_1 ln\theta + b_2 ln(1-\theta)$]

=b$_1 \frac{d}{d\theta}\ln\theta$+b$_2 \frac{d}{d\theta}$ln(1-$\theta$)

=b$_1 \frac{1}{\theta}$+$b_2 \frac{-1}{1-\theta}$

Nota: Mai sus am folosit derivatele:

$\frac{d}{dx}$lnx=$\frac{1}{x}$

$\frac{d}{dx}$ln g(x)=$\frac{1}{g(x)} \frac{d}{dx}$ g(x)

Observam ca am obţinut acelaşi rezultat ca mai sus deci algortimul intuitiv de mai sus este echivalent cu estimarea în sens MLE.

2.2.1.2 Estimarea MAP

Estimatorul MAP ne permite sa folosim cunoştinţe anterioare despre distribuţia a priorii p() (folosim aceeaşi notaţie ca mai sus). Aceste cunoştinţe pot rezulta din experienta noastră, din domeniul de specialitate al problemei sau pot fi intuitive. Astfel, estimarea MAP se va folosi de probabilitatea a posteriorii:

$$\hat{p} = argmax_p P(D \mid p) \tag{5}$$

Pentru a ajunge la forma finala la formula de mai sus am folosit formula lui Bayes, astfel:

$$P(p \mid D) = \frac{P(D \mid p)P(p)}{P(D)} \tag{6}$$

Numitorul a fost mai apoi ignorat întrucât nu depinde de p. In general, atunci când ne confruntam cu rezultate consecutive ale unei variabile aleatoare Bernoulli, datele urmează o distribuţie Beta. Prin urmare putem scrie P(p) astfel:

$$P(p) = Beta(b_1, b_2) = \frac{p^{b_1-1}(1-p)^{b_0-1}}{B(b_0, b_1)} \tag{7}$$

Unde $B(b_1, b_2)$ este un factor de normalizare pe care il vom ignora in continuare întrucât nu depinde de p. Cu aceasta noua informaţie putem sa rescriem prima ecuaţie astfel:

$$\begin{aligned} p^{MAP} &= argmax_p P(D \mid p)P(p) \\ &= argmax_p p^{a_1}(1-p)^{a_0} \frac{p^{b_1-1}(1-p)^{b_0-1}}{B(b_0, b_1)} \\ &= argmax_p \frac{p^{a_1+b_1-1}(1-p)^{a_0+b_0-1}}{B(b_0, b_1)} \\ &= argmax_p p^{a_1+b_1-1}(1-p)^{a_0+b_0-1} \end{aligned}$$

Observam ca pentru a afla valoarea p care maximizează funcţia de mai sus ne putem folosi de rezultatul obţinut la Estimarea MLE, înlocuind $b_1 = a_1 + b_1 - 1, b_2 = a_2 + b_2 - 1$. In final obţinem:

$$p^{MAP} = argmax_p P(D \mid p)P(p) = \frac{(a_1 + b_1 - 1)}{(a_1 + b_1 - 1) + (a_0 + b_0 - 1)} \tag{8}$$

,acelaşi rezultat pe care l-am obţinut şi în algoritmul intuitiv iniţial dacă facem substituia $b_1 = \beta_1 - 1$ si $b_2 = \beta_2 - 1$. Deci, folosind estimarea în sens MAP - mai exact dorind sa găsim valoarea lui $P(\theta)$) care maximizează $P(\theta \mid D)$ - am obţinut acelaşi rezultat

2.3 Algoritmul Bayes Naiv

2.3.1 Noţiunea de Independenţa Condiţională

Spunem ca doua variabile aleatoare X, Y sunt independente conditional în raport cu o a treia variabila aleatoare Z dacă următoarea egalitate este îndeplinita:

$$P(X,Y \mid Z) = P(X \mid Z)P(Y \mid Z) \tag{9}$$

2.3.2 Introducere Bayes Naiv

Fiind dat un set de date de antrenare X=$X_1, X_2, X_3, .., X_n$ şi etichetele corespunzătoare reprezentarea de variabila discreta Y=Y_1, Y_2, .. Y_n, algoritmul Bayes Naiv de foloseşte de formula lui Bayes enunţată la 2.1.1 si de presupunerea de independenta conditionala a datelor de antrenare pentru a estima eticheta Yk a unei noi instante X_k=x_1,x_2,... Pentru a face acest lucru, algoritmul va învaţă o distribuţie asupra probabilităţilor pentru fiecare valoare posibila a etichetei Y_k în raport cu noua instanta de clasificat X_k=x_1,x_2,.. şi în final va alege cea mai probabila eticheta.

Conform 2.2.1, putem scrie probabilitatea ca Y_k sa aiba valoarea y_k pentru instanta X_k=x_1,x_2,x_3 astfel:

$$P(Y_k = y_k \mid x_1, x_2..) = \frac{P(Y_k = y_k)P(x_1, x_2, x_3.. \mid Y = y_k)}{\sum_j P(Y_k = y_j)P(x_1, x_2, x_3.. \mid Y_k = y_j)} \tag{10}$$

Nota: Pentru exprimarea numitorului P(x1, x2, x3..) s-a folosit Formula Probabilitatii Totale

Presupunând ca x_1, x_2, x_3 sunt independente conditional în raport cu y_k vom rescrie ecuaţia de mai sus astfel: P($Y_k = y_k \mid x_1, x_2..$) = $\frac{P(Y_k=y_k)\prod_i P(x_i|Y=y_k)}{\sum_j P(Y_k=y_j)\prod_i P(x_i|Y_j)}$

în final, pentru a afla eticheta cea mai probabila trebuie doar sa aflam maximul dintre toate valorile posibile pentru Y:

$$Y_k = argmax_k P(Y_k = y_k \mid x_1, x_2, ..) = argmax_k \frac{P(Y_k = y_k)\prod_i P(x_i \mid Y = y_k)}{\sum_j P(Y_k = y_j)\prod_i P(x_i \mid Y_j)} \tag{11}$$

întrucât numitorul nu depinde de y_k îl putem ignora:

$$Y_k = argmax_k P(Y_k = y_k) \prod_i P(x_i \mid Y = y_k) \tag{12}$$

2.3.3 Relaţia de echivalenţa in exprimare dintre Bayes Naiv si Regresia Logistica pentru variabile discrete

In continuare datele de intrare vor avea aceeasi forma descrisa mai sus. In plus, vom presupune ca variabilele X_i sunt boolene. Vom demonstra ca distributia conditionala $P(Y \mid X)$ are forma functiei logisitice de parametru w, descrisa mai in detaliu in sectiunea 3, si anume:

$$P(Y = 1 \mid X) = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i} \tag{13}$$

Vom introduce urmatoarele notaţii: $P(X_i = 1 \mid Y = 1) = \theta_{i1} \Rightarrow P(X_i = 0 \mid Y = 1) = 1 - \theta_{i1}$, $P(X_i = 1 \mid Y = 0) = \theta_{i0} \Rightarrow P(X_i = 0 \mid Y = 0) = 1 - \theta_{i0}$. Facand aceste notatii putem scrie acum probabilitatea $P(X_i \mid Y = 1)$ intr-un mod compact astfel:

$$P(X_i \mid Y = 1) = \theta_{i1}^{X_i}(1 - \theta_{i1})^{1 - X_i} \tag{14}$$

Formula lui Bayes ne permite sa exprimam $P(Y = 1 \mid X)$ astfel:

$$P(Y = 1 \mid X) = \frac{P(Y = 1)P(X \mid Y = 1)}{P(Y = 1)P(X \mid Y = 1) + P(Y = 0)P(X \mid Y = 0)} \tag{15}$$

Scotand factor comun $P(Y = 1)P(X \mid Y = 1)$ la numitor si facand simplificarile obtinem:

$$P(Y = 1 \mid X) = \frac{1}{1 + \frac{P(Y=0)P(X=0)}{P(Y=1)P(X|Y=1)}} \equiv \frac{1}{1 + exp(ln(\frac{P(Y=0)P(X=0)}{P(Y=1)P(X|Y=1)))}} \tag{16}$$

Vom nota probabilităţile a priorii $P(Y = 1)$ şi $P(Y = 0)$ cu *pi* respectiv $1 - \pi$. De asemenea, întrucât Bayes Naiv lucrează cu presupunerea că pentru un Y dat, $\forall$ i şi j $\neq$ i, $X_{şi}$ sunt X_j independente mai

putem rescrie ecuaţia de sus astfel:

$$P(Y=1 \mid X) = \frac{1}{1+exp(ln(\frac{P(Y=0)}{P(Y=1)} + \sum_i ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})} \equiv \frac{1}{1+exp(ln(\frac{1-\pi}{\pi} + \sum_i ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})} \quad (17)$$

In continuare, conform (14), vom scrie $P(X \mid Y=1)$ ca $\theta_{i1}^{X_i}(1-\theta_{i1})^{1-X_i}$ si $P(X_i \mid Y=0)$ ca $\theta_{i0}^{X_i}(1-\theta_{i0})^{1-X_i}$. Ecuatia devine:

$$P(Y=1 \mid X) = \frac{1}{1+exp(ln\frac{1-\pi}{\pi} + \sum_{i=1}^{d} ln\frac{1-\theta_{i0}}{1-\theta_{i1}} + \sum_{i=1}^{d} X_i(ln\frac{\theta_{i0}}{\theta_{i1}} - ln\frac{1-\theta_{i0}}{1-\theta_{i1}}))} \quad (18)$$

Pentru ca aceasta ecuatie sa aiba forma regresiei logistice, vom alege:

$$w_0 = ln\frac{1-\pi}{\pi} + \sum_{i=1}^{d} ln\frac{1-\theta_{i0}}{1-\theta_{i1}} siw_i = ln\frac{\theta_{i0}}{\theta_{i1}} - ln\frac{1-\theta_{i0}}{1-\theta_{i1}} \quad (19)$$

2.3.4 Algoritmul Bayes Naiv pentru variabile continue

Desi formula pentru prezicere a etichetelor folosind algoritmul Bayes Naiv este aceeasi indifirent daca lucram cu date cu valori discrete sau continue, ceea ce diera este modul in care calculam valoarea $P(X_i \mid Y)$.

În cazul când avem valori discrete, valoarea $P(X_i midY)$ se obţine imediat folosind estimarea MAP sau MLE. În cazul valorilor continue, cea mai folosită modalitate de a reprezenta distribuţia $P(X_i \mid Y)$ este să se folosească presupunarea că pt fiecare valoare discretă y_k, disributia lui $X_u na$ este Gaussiană şi anumită este definită de o şi medie şi deviaţie standard specifice lui X_a antrena y_k. În continuare, aceste valori le vom reprezenta astfel:

$$\mu_{ik} = E[X_i \mid Y = y_k] \quad (20)$$

$$\sigma_{ik}^2 = E[(X_i - \mu_{ik})^2 \mid Y = y_k] \quad (21)$$

De asemenea, trebuie sa estimam si probabilitatea a priorii a lui Y:

$$\pi_k = P(Y_k = y) \quad (22)$$

Folosind aceste valori putem scrie $P(X_i \mid Y)$ astfel:

$$P(X_i \mid Y = y_k) = \frac{1}{\sqrt{2\pi}\sqrt{\sigma_{ik}^2}} e^{\frac{-(X_i-\mu)^2}{2\sigma_2}} \tag{23}$$

Pentru a estima media si deviatia standard putem folosi MLE si obtinem:

$$\hat{\mu_{ik}} = \frac{1}{\sum_j \sigma(Y_j = y_k)} \sum_j X_i^j \sigma(Y^j = y_k) \tag{24}$$

$$\hat{\sigma_{ik}^2} = \frac{1}{\sum_j \sigma(Y_j = y_k)} \sum_j (X_i^j - \mu_{ik})^2 \sigma(\hat{Y}^j = y_k) \tag{25}$$

2.3.5 Bayes Naiv Gaussian

Folosind aceeasi presupunere ca mai sus si anume ca pentru fiecare X_i, $P(X_i \mid Y = y_k)$ are o distributie Gaussiana data de parametrii μ_{ik} si σ_{ik}^2, vom exprima acum termenul $P(Y \mid X)$. In plus fata de mai sus, mai facem presupunerea ca Y este o variabila booleana, avand o distribuite Bernoulli data de parametrul $\pi = P(Y = 1)$. Vom observa ca se va obtine aceeasi forma pe care o folosim la alt algoritm de clasificare: Regresia Logistica (vezi sectiunea 3).

Formula lui Bayes ne permite sa exprimam $P(Y = 1 \mid X)$ astfel:

$$P(Y = 1 \mid X) = \frac{P(Y = 1)P(X \mid Y = 1)}{P(Y = 1)P(X \mid Y = 1) + P(Y = 0)P(X \mid Y = 0)} \tag{26}$$

Scoatand factor comun $P(Y = 1)P(X \mid Y = 1)$ la numitor si facand simplificarile obtinem:

$$P(Y = 1 \mid X) = \frac{1}{1 + \frac{P(Y=0)P(X=0)}{P(Y=1)P(X|Y=1)}} \equiv \frac{1}{1 + exp(ln(\frac{P(Y=0)P(X=0)}{P(Y=1)P(X|Y=1)))}} \tag{27}$$

Întrucât Bayes Naiv lucrează cu presupunerea că pentru un Y dat, $\forall$ i şi j $\neq$ i, $X_{şi}$ sunt X_j independente mai putem rescrie ecuaţia de sus astfel:

$$P(Y = 1 \mid X) = \frac{1}{1 + exp(ln(\frac{P(Y=0)}{P(Y=1)} + \sum_i ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})} \equiv \frac{1}{1 + exp(ln(\frac{1-\pi}{\pi} + \sum_i ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)})} \tag{28}$$

Utilizând (23) şi fâcând calculele putem scrie:

$$\sum_i ln\frac{P(X_i=0)}{P(X_i \mid Y=1)} = \sum_i (X_i\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}) \tag{29}$$

$(28), (29) \implies$

$$P(Y=1 \mid X) = \frac{1}{1+exp(ln\frac{1-\pi}{\pi} + \sum_i (X_i\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2})} \tag{30}$$

Notam $ln\frac{1-\pi}{\pi} + \sum_i (\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}))$ cu w_0, respectiv $\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}$ cu w_i $\forall$i ≥ 1 si obtinem:

$$P(Y=1 \mid X) = \frac{1}{1+exp(w_0 + \sum_i w_i X_i)} \tag{31}$$

2.4 Concluzie capitol

În acest capitol am făcut o prezentare scurtă a algoritmului Bayes Naiv şi modul în care acesta lucrează atunci când avem date discrete şi continue. De asemenea, am arătat cum formă folosită de Regresia Logisitica poate fi echivalentă cu cea folosită de Algoritmul Bayes Naiv. Un alt concept introdus a fost o variaţie al algoritmului Bayes Naiv-algoritmul Bayes Naiv Gaussian-care presupune o distribuţie Gaussiană asupra datelor de antrenare.

3 Modelul Discriminativ

3.1 Introducere capitol

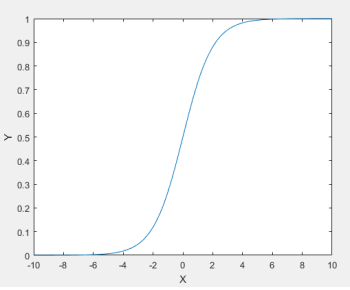
Comparativ cu modelul generativ, un model discriminativ este un model care nu se interesează de modul în care datele au fost generate. Acesta din urmă încearcă să înveţe nişte graniţe de decizie pe care le va folosi mai apoi pentru clasificare unei noi instanţe.

În exemplul de clasificare între motociclete şi maşini, un algoritm discriminativ va învaţă întâi diferenţele dintre o motocicletă şi o maşină şi folosindu-se de aceste cunoştinţe va determinaă clasă unui nou obiect.

3.2 Regresia Logistica

3.2.1 Introducere Regresie Logistica

Figura 1: Funcţia sigmoid



Ca în orice problemă de clasificare, datele de intrare sunt n instanţe de antrenament $X_1,..X_n$, $X_i = x_1,..x_d \forall i$. Fiecărei instanţe îi corespunde o etichetă Y_i. Regresia încearcă să să înveţe o funcţie care a aproximeze distribuţia $P(Y \mid X)$. Pentru că face acest lucru, acest algoritm presupune aproximată $P(Y \mid X)$ poate fi ai cu o funcţie sigmoidala iar apoi estimează parametrii acestei funcţii direct din datele de antrenare.

$$\sigma(x) \stackrel{\text{def}}{=} \frac{1}{1+e^{-z}} \tag{32}$$

În cazul în care Y este o variabilă booleană, regresia logistică foloseşte următoarea ecuaţie pentru a exprima $P(Y = 1 \mid X)$:

$$P(Y = 1 \mid X) = \prod_i \sigma(w_i X_i) \tag{33}$$

Nota: din (33) se obţine imediat $P(Y = 0 \mid X)$ folosindu-ne de faptul ca suma acestor 2 probabilităţi

este 1.

În continuare vom arată modul în care regresia logisiticâ alege valori optime pentru parametrii w_i cum se aleg regulile de actualizare pentru aceşti parametri.

3.2.2 Estimarea parametrilor

Regresia Logistică alege parametrii w_i urmărind să maximiezeze log-verosimilitatea condiţională a întregului set de date de antrenament. Verosimilitatea conditionala este probabilitatea sa observam etichetele Y, fiind dat setul de date X. Notam cu W vectorul $w_0, w_1, w_n...$ Ne dorim sa alegem W astfel incat:

$$W \leftarrow argmax_W \prod_l P(Y^l \mid X^l, W) \tag{34}$$

Observăm că putem scrie probabilitatea $P(Y = y \mid X = x)$ astfel:

$$P(Y = y \mid X = x) = \sigma(wx)^y [1 - \sigma(wx)]^{1-y} \tag{35}$$

Folosind aceasat ecuaţie verosimilitatea condiţională devine:

$$\prod_l P(Y = y^{(l)} \mid X = x^{(l)}) = \prod_l (\sigma(wx^{(l)})^{y(l)} [1 - \sigma(wx^{(l)})]^{1-y^{(l)}}) \tag{36}$$

Aplicând logaritmul şi notând log-verosimilitatea condiţională cu $l(W)$ obţinem:

$$l(W) = \sum_l (y^{(l)} ln\sigma(wx^{(l)}) + (1 - y^{(l)}) ln[1 - \sigma(wx^{(l)})]) \tag{37}$$

Întrucât nu există o formulă directă pentru determinarea acelor valori ale lui W care să maximizeze log-verosimilitatea condiţională a datelor, vom determină aceste valori folosind o metodă de optimizare. Există mai multe metode de optimizare dar în continuare vom alege o singură metodă: gradientul ascendent. Ideea acestei metode este să pornim cu nişte valorilor iniţiale pentru parametrii w (valori alese aleator sau în urmă unor cunostine anterioare) şi apoi să facem în mod continuu păşi mici în direcţia gradientului funcţiei l(W) (vectorul de derivate parţiale). Astfel, vom ajunge într-un final în punctul de maxim global.

3.2.3 Actualizarea valorilor w_i

Pentru a maximiza funcţia de log verosimilitate condiţională, la fiecare pas optimizăm parametrul w astfel:

$$W_j^{nou} = W_j^{vechi} + \alpha \frac{\delta}{\delta W_j^{vechi}} L(W^{vechi}) \tag{38}$$

unde α este marimea pasului, numita rata de invatare.

În continuare vom calcula valoarea derivatei funcţie de log-verosimilitate condiţională. Aceasta se va calcula ca o sumă de derivate parţiale. Pentru o instata (x,y) derivata se calculează astfel:

$$\begin{aligned}
\frac{\delta}{\delta W_j} L(w_j) &= \frac{\delta}{\delta W_j} y ln\sigma(wx) + \frac{\delta}{\delta W_j}(1-y) ln[1-\sigma(wx)] \\
&= [\frac{y}{\sigma(wx)} - \frac{1-y}{1-\sigma(wx)}] \frac{\delta}{\delta w_j} \sigma(wx) \\
&= [\frac{y}{\sigma(wx)} - \frac{1-y}{1-\sigma(wx)}] \sigma(wx)[1-\sigma(wx)] x_j \\
&= \frac{y-\sigma(wx)}{\sigma(wx)[1-\sigma(wx)]} \sigma(wx)[1-\sigma(wx)] x_j \\
&= [y-\sigma(wx)] x_j
\end{aligned}$$

Însumând aceşti termeni pentru fiecare instanţă de antrenare obţinem derivata log verosimilităţii condiţionale a întregului se de date:

$$\frac{\delta}{\delta w_j} L(w) = \sum_i [y^{(i)} - \sigma(wx^{(i)})] x_j^{(i)} \tag{39}$$

Notă: Mai sus ne-am folosit de următoarea proprietate pentru derivata funcţiei logistice σ în raport cu argumentul ei:

$$\frac{\delta}{\delta w_j} \sigma(z) = \sigma(z)[1-\sigma(z)] \forall z \in R \tag{40}$$

3.2.4 Teorema de convergenta pentru Regresia Logistica

Pentru a demonstra că algoritmul de optimizare al parametrului W ajunge într-un maxim global vom demonstra că funcţia de log-verosimilitate condiţională a datelor de antrenare este concava şi deci nu are alt maxim local. Vom scrie întâi matricea gradient H a funcţiei L(w) formată din pătratele derivatelor de ordin 2 ale paramtrului W. Vom arăta apoi că pentru orice vector $z =$

$(z_1, z_2...)$ următoarea ecuaţie este adevărată: $z^T Hz \leq 0$. Acest lucru înseamnă că matricea H este negativ semi-definită şi deci funcţia L(w) este concavă.
Folosindu-ne de ecuaţiile $(30), (31)$ putem exprima termenii matricii H astfel:

$$H_{kl} = \frac{\delta^2 L(w)}{\delta w_k \delta w_l} = -\sum_i \frac{\delta \sigma(wx^{(i)})}{\delta w_l} x_k^{(i)} = -\sum_i \sigma(wx^{(i)})(1 - \sigma(wx^{(i)}) x_l^{(i)} x_k^{(i)} \tag{41}$$

tiind că $X = xx^T \iff X_{ij} = x_i x_j$ matricea H devine:

$$H = -\sum_i \sigma(wx^{(i)})(1 - \sigma(wx^{(i)}) x^{(i)} x^{(i)^T} \tag{42}$$

Pentru a arata ca H este negativ semi-definita, vom arata ca $z^T Hz \leq 0 \forall z$

$$\begin{aligned} z^T Hz = & \quad -z^T (\sum_{i=1}^{m} \sigma(x^{(i)})(1 - \sigma(x^{(i)})) x^{(i)} x^{(i)T}) z \\ = & \quad -\sum_{i=1}^{m} \sigma(x^{(i)})(1 - \sigma(x^{(i)})) z^T x^{(i)} x^{(i)^T} z \\ = & \quad -\sum_{i=1}^{m} \sigma(x^{(i)})(1 - \sigma(x^{(i)}))(z^T x^{(i)})^2 \leq 0 \end{aligned}$$

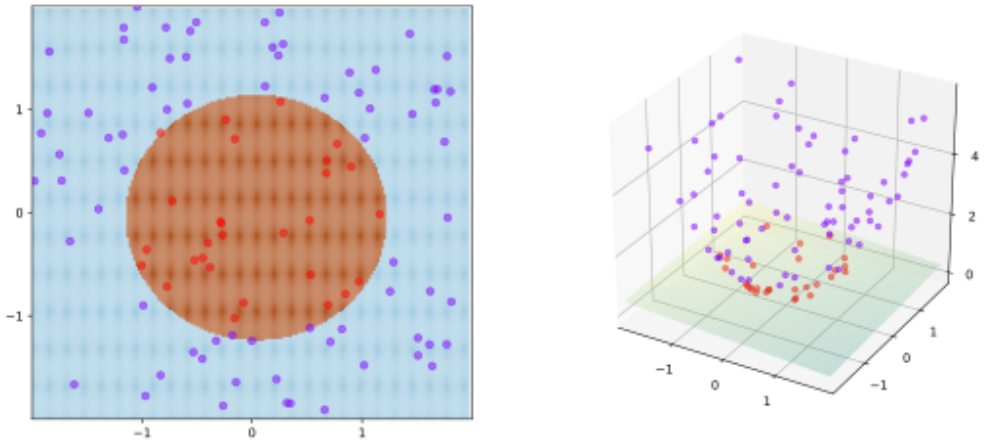
Notă: Ultima inegalitate este adevărată întrucât $0 \leq \sigma(x^{(i)}) \leq 1$ si $(z^T x^{(i)})^2 \geq 0$

3.2.5 Regularizarea regresiei logistice

Atunci când datele de antrenare au multe atribute sau numărul de insante pentru antrenare este redus o problemă care poate apărea folosind regresia logistică este fenomenul de ”overfitting”. Astfel, algoritmul va avea o acurateţe mare pentru datele de antrenare dar una redusă pentru datele de testare. O modalitate de a evită acest lucru este regularizarea regresiei logistice. Mai exact, vom creă o versiune modificată a funcţiei de log verosimilitate condiţională care va penaliza valorile mari ale parametrului W.

$$W \leftarrow argmax_W \sum_l ln(P(Y^l \mid X^l, W) - \lambda \mid\mid W \mid\mid^2 \tag{43}$$

Figura 2: Maparea datelor initiale intr-un spatiu 3d unde se observa ca acestea pot fi separate de un plan. Pentru mapare s-a folosit functia Kernel K(x, y) = xy + x^2y^2. (Figura produsa de Shiyu Ji - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60458994)



În funcţie de modul în care este penalizat paramtrul W, există mai multe tipuri de regularizări. Cele mai întâlnite tipuri sunt regularizarea L1 şi regularizarea L2. Un model care foloseşte regularizarea L1 se numeşte Regresie Lasso iar un model care foloseşte regularizarea L2 se va numi Regresie Ridge. Regularizarea L1 penalizează funcţia de verosimilitate condiţională cu $\lambda \sum_i \mid w_i \mid$ iar regularizarea L2 cu $\lambda \sum_i w_i^2$. Deşi ambele au ca scop evitarea producerii fenomenului de overfitting există câteva diferenţe care ne vor ajută să alegem tipul de regularizare potrivit. Regularizarea L1 este ineficienţa computaţional atunci când nu avem multe elemente de 0. De asemenea această va produce multe dintre elementele lui W să fie 0, un lucru avantajos atunci când ne dorim să facem şi selectare de trsaturi pentru setul nostru de date.

3.2.6 Regresia Logistica Kernelizata

Atunci cand ne confruntam cu un set de date care nu sunt separabile liniar putem incerca sa transpunem datele intr-un spatiu de dimensiune mai mare in care datele for deveni separabile liniar.

In general, kernelizarea presupune gasirea unei functii de mapare $f : X \rightarrow Z$, unde X este spatiul initial al datelor iar Z este un spatiu de dimensiune mai mare, astfel incat calculele in Z sa se foloseasca doar de produsul scalar al vectorilor si exista o functie K, numita kernel, astfel incat produsul scalar dintre $f(x_i)$ si $f(x_j)$ sa fie $K(x_i, x_j)$.

Notând cu n dimensiunea spaţiului de date X şi cu m dimemnsiunea spaţiului Z în care ne dorim să proiectăm datele, forma generală a regresiei logistice va deveni:

$$P(Y = 1 \mid f(X)) = g(w_0 + \sum_{i=1}^{m} w_i f(X)_i) \tag{44}$$

unde g este functia sigmoid

Verosimilitatea parametrului W este

$$l(W) = \sum_{l=1}^{R} Y^l (w_0 + \sum_{j=1}^{R} w_j(X_j, X)) - ln(1 + exp(w_0 + \sum_{j=1}^{R} w_j(X_j, X))) \tag{45}$$

Pentru a obţine regula de actualizare a parametrului W, calculăm derivată ecuaţiei de mai sus:

$$\frac{\delta(l(W))}{\delta w_i} = \sum_{l=1}^{R} (Y^l - \frac{exp(w_0 + \sum_{j=1}^{R} w_j(X_j, X))}{1 + exp(w_0 + \sum_{j=1}^{R} w_j(X_j, X))}) K(X_i, X) \tag{46}$$

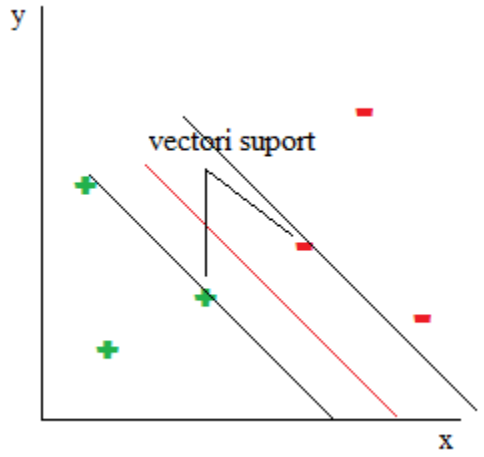
Regula de actualizare:

$$W^{nou} = W^{vechi} + \alpha \frac{\delta(l(W))}{\delta W} \tag{47}$$

3.3 Maşini cu Vector Suport

Un alt algoritm discriminativ care a devenit foarte popular in ultima vreme sunt Masinile cu vector suport (SVM). Masinile cu vector suport au ca scop desenarea unui hiperplan care imparte cel mai bine setul de date in clasele in care se doreste a fi facuta clasificarea. .

Figura 3: SVM



Consideram instantele $x_1, x_2, ..x_m \in R^d$ si etichetele corespunzatoare $y_1, y_2..y_m \in -1, 1$

Vectorii suport vor fi punctele care sunt cel mai aproape de linia care separă setul de date. Distanţa dintre un vector suport şi separator poartă numele de margine.În cazul în care datele de antrenare sunt eparabile liniar, scopul algoritmului este să găsească o margine cât mai mare. Găsirea acestei margini este o problemă de optimizare şi poartă numele de ”Problemă SVM cu margine hard”. Odată găsită această margine, folosind figură 3, regulă de decizie este următoarea: noi puncte vor fi clasificate ’+’ dacă vor fi la stânga suprafeţei de decizie, sau cu ’-’ în caz contrar.

Matematic, aceasta problema se exprima astfel:

$$min_{w,w_0} \frac{1}{2} \mid\mid w \mid\mid^2 a.i.(wx_i + w_0)y_i \geq 1 \forall i = 1..m \tag{48}$$

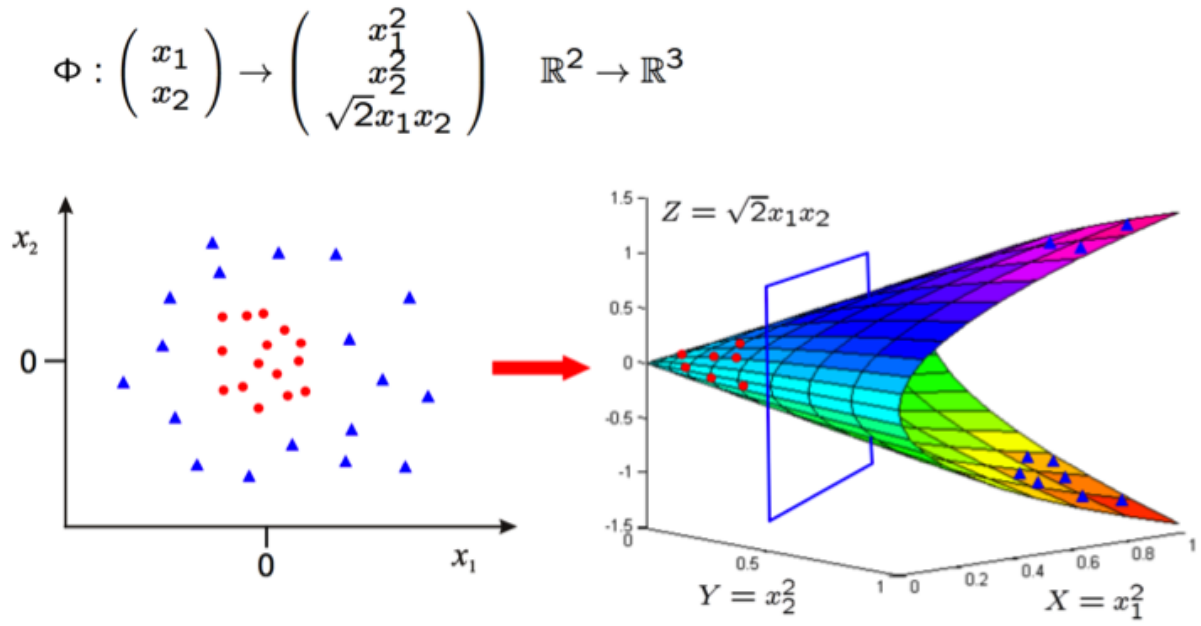
unde $w \in R^d$ si $w_0 \in R$ În urma rezolvării acestei probleme se va obţine un model liniar, de forma $y(x) = wx + w_0$, ce va servi ulterior pentru clasificare conform functiei sign(y(x)), echivalenta cu regula de decizie mentionata anterior.

În practică însă, datele de antrenare nu sunt întotdeauna perfect separabile liniar. În acest caz folosirea marginii hard nu va mai da rezultate aşa că se va folosi ”marginea soft”. Problemă SVM cu margine soft va conţine în plus un parametru de regularizare care va specifică numărul de date clasificate greşit pe care îl permite. Atunci când parametrul de regularizare este mic, clasficatorul va permite că un mic număr din datele de antrenare să fie clasificate greşit. Deşi pe datele

de antrenare clasificatorul va avea o performanţă destul de bună, acesta nu va generaliza foarte bine, producându-se astfel fenomenul de 'overfitting'. Atunci când termenul de regularizare va fi mai mare, clasificatorul va generaliza mai bine. Totuşi, valori prea mari nu sunt indicate întrucât performanţă clasificatorului poate scădea destul de mult.

Atunci când datele nu sunt deloc separabile liniar, se va încerca transpunerea datelor într-un spaţiu de dimensiune mai mare. Acest lucru este cunoscut sub numele de 'kernelizare'. O funcţie kernel este o funcţie care cuantifica similarităţile dintre observaţii.

Figura 4: Datele sunt separabile liniar atunci cand le transpunem intr-un plan 3D ©towardsdatascience.com



3.4 Concluzie capitol

În acest capitol am descris modul în care un model discriminativ clasifică un set de date, prezentând în special 2 algoritmi: Regresia Loistica şi Maşinile cu Vector Suport. Pentru regresia loistica,

care este considerat perechea discriminativa a algoritmului Bayes Naiv, am arătat modul în care se estimează parametrii, precum şi 2 metode de îmbunătăţire: regularizarea şi kernelizarea.

4 Exemple Aplicative

4.1 Aplicare Regresie Logistica

Pentru început vom folosi Regresia Logistică pentru a rezolva o problemă de clasificare binară. Vom aplica algoritmul pe un set de date preluat din baza de date ”UCI Machine Learning Repository”. Setul de date cuprinde informaţii despre comunităţile din America precum şi o clasificare a acestor comunităţi dacă sunt sau nu sigure considerând rata infracţiunilor din acea zonă. În total sunt 1994 de observaţii, fiecare având 128 de atribute.

După ce am rulat algoritmul am obţinut o acurateţe de 83.4% . După cum observăm în prima figură, unde am afişat valorile în medie a parametrilor W, raportat la cantitatea în care se găsesc aceştia, obţinem valori foarte mari pentru W. Acest lucru nu este de dorit întrucât asta implică o şansa mai mare în apariţia fenomenului de overfitting. Pentru a evita acest lucru ne vom folosi de regularizarea L1 şi L2 a regresiei logistice.

Figura 1

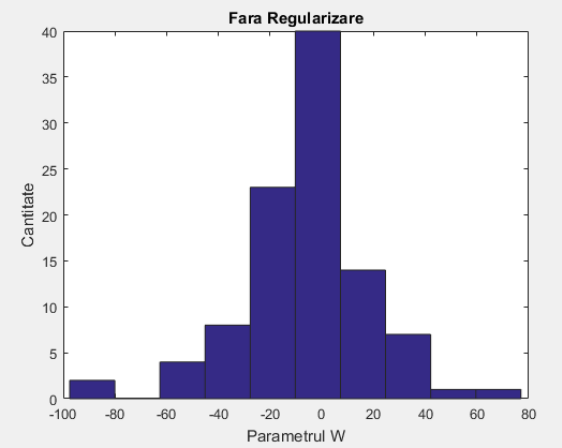
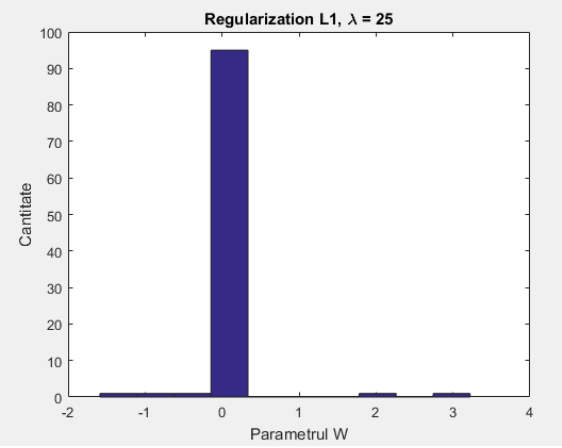


Figura 2 : Regularizare L1

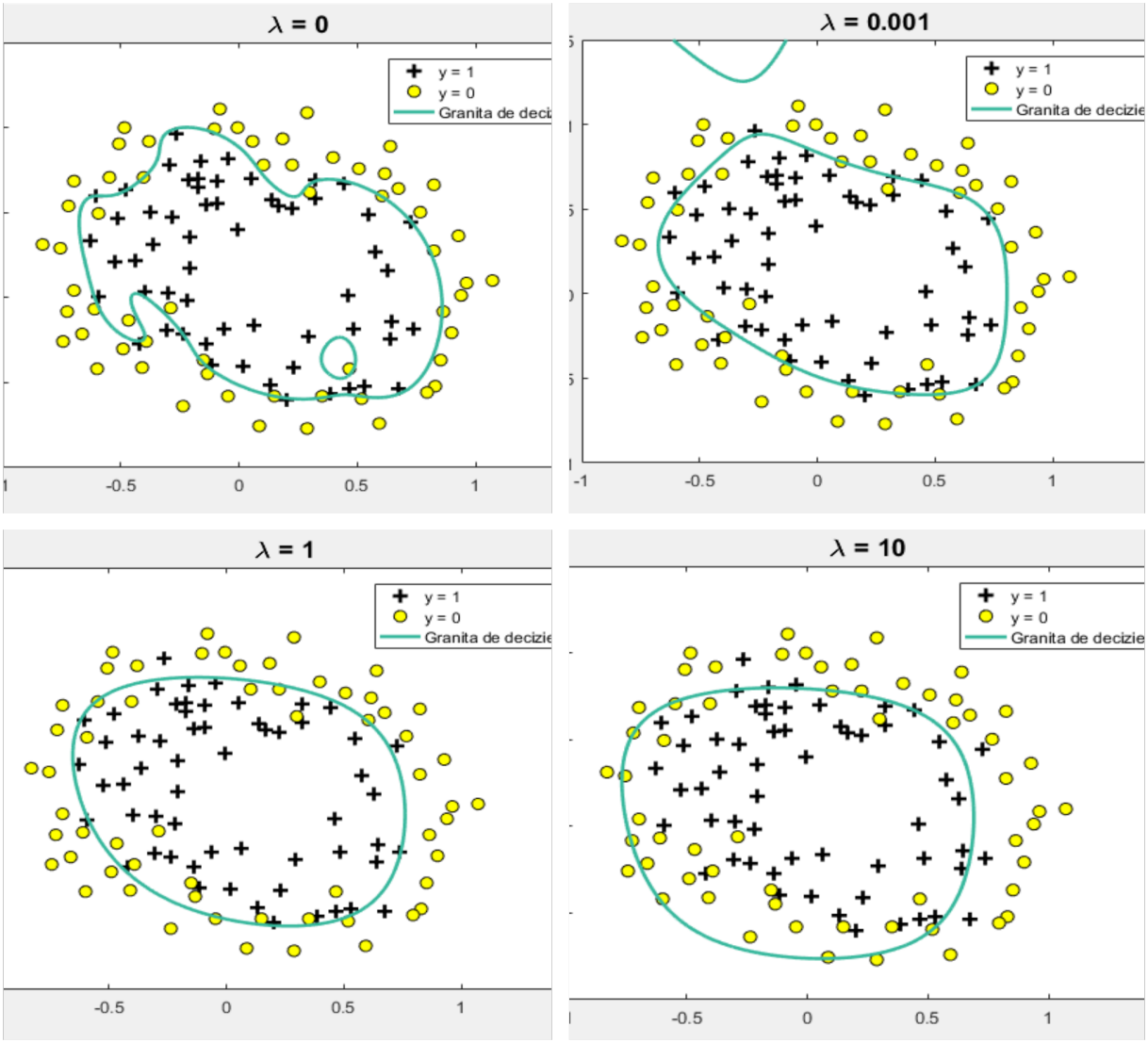


După cum observăm în figura a două, aplicarea regularizării L1 a transformat mulţi parametri în 0, ceea ce ne ajută atunci când spaţiul de date de antrenare este de o dimensiune foarte mare, întrucât acest lucru ne permite să facem selectare a atributelor. În ceea ce priveşte acurateţea la cros validare după aplcarea regularizării L1, aceasta nu a crescut foarte mult, procentul exact fiind: 83.5%.

4.1.1 Vizualizarea graniţei de decizie

Pentru a vizualiza mai bine modul în care regularizarea regresiei logistice influenţează graniţă de decizie ne vom folosi în continuare de un set de date preluat din cursul de Machine Learning al profesorului Andrew Ng. Datele de antrenare sunt din spaţiul R^2 şi avem şi o clasificare booleană a acestor date. Mai jos am prezentat distribuţia iniţială a datelor precum şi modul în care graniţa de decizie se modifică în funcţie de mărimea parametrului de regularizare λ

Figura 5: Graniţa de decizie pentru diferite valori λ



După cum observăm în figura de mai sus, atunci când nu avem deloc regularizare algoritmul tinde să se modeleze cât mai mult pe datele de antrenare, ceea ce va determina o eroare mai mare

pe datele de test. În acelaşi timp, alegerea unei valori prea mari pentru λ poate diminua cu mult acurateţea clasificatorului atât pe datele de antrenare cât şi pe ele de test.

4.1.2 Regresia Logistica MultiClass

Regresia logistică poate fi aplicată şi atunci când avem de rezolvat o problemă de clasificare cu mai multe clase. Pentru următoarea aplicaţie am antrenat regresia logistică pentru a distinge între imagini alb-negru cu cifre. Pentru fiecare cifră din intervalul [0-9] am avut 600 de imagini de antrenare şi 500 de testare. Fiecare imagine are dimensiunea de 16*16 px şi e reprezentată de un vector de dimensiune 256 în care fiecare element este 0 dacă pixelul respectiv este negru sau 1 dacă pixelul este alb.

După ce am aplicat regresia logistică am obţinut o acurateţe de 91.44 %. Aplicând regularizarea L2, am obţinut o acurateţe de 91.92% atunci când am ales $\lambda = 10$.

4.1.3 Alte Observatii

Figura 6: Marimea setului de date de antrenare vs Eroare la testare si antrenare ©CMU, T. Mitchel

Când setul de date de antrenare este mic, regresia logistică poate de obicei să clasifice perfect datele de antrenare întrucât variant datelor este foarte redusă. Acesta este motivul pentru care eroarea la antrenare este aproape 0. Acest model nu are însă capabilitatea de generalizare, întrucât estimările parametrului w sunt bazate pe un subset de date care nu este reprezentativ pentru întregul set de date. Acest lucru este cunoscut sub numele de overfitting. Pe măsură ce avem mai multe date de antrenare, variantă datelor creşte şi modelul nu mai este capabil de obicei să prezică correct toate datele. În acelaşi timp, mai multe date de antrenare oferă modeului nostrum o viziune mai largă asupra întregului set de date ceea ce îi permitesa aibă o acurateţe mai mare în estimarea parametrului what. Acest lucru duce la o generalizare mai bun, deci o eroare la testare mai mică.

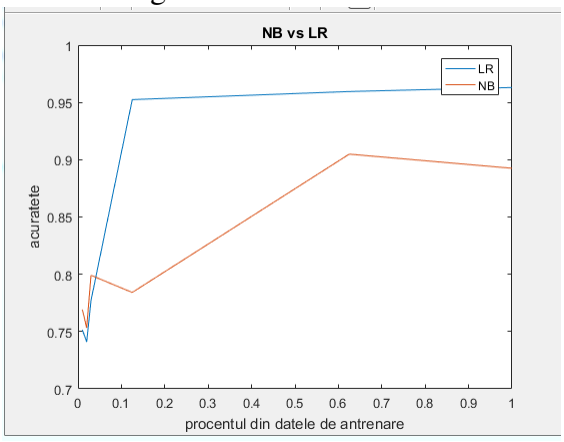
4.2 Aplicare Bayes Naiv vs Regresia Logistica

Pentru a arăta diferenţa dintre Bayes Naiv şi Regresia Logisitca am lucrat pe un set de date din baza de date ”UCI Machine Learning Repository”. Setul de date cuprinde informaţii despre pacienţi diagnozicati cu cancer la sân precum şi o clasificare a tumorilor canceroase ca fiind inofensive sau periculoase. În total setul de date are 699 de observaţii, din care am eliminat 6 pentru care aveam informaţii lipsă. Fiecare observaţie are 10 atribute.

Pentru a compara cei doi algoritmi am folosit metoda 10-fold cross validation în care împărţim setul de date în 10 părţi egale, antrenăm algoritmul pe 9 subseturi de date şi testăm pe un subset. Acest lucru îl facem pentru fiecare din cele 10 susbseturi. În urma cross validării am obţinut următoarele rezultate:

- Bayes Naiv: 0.9103 acuratete
- Regresia Logistica: 0.9853 acuratete

Figura 7: NB vs LR



În lucrarea ”On discriminative and generative classifiers”, profesorul Andrew Ng. spune ca ”Pe masura ce marimea setului de date de antrenament creste, initial algorimult Bayes Naiv va da rezultate mai bune dar in final Regresia Logisitca va ajunge la aceeasi performanta iar mai apoi probabil va avea chiar rezultate mai bune”. Pentru a testa acest lucru am impartit datele in 3 sferutri pentru antrenare si un sfert pentru testare. Apoi, din datele de antrenare am antrenat algoritmul doar cu 1%, 2%, 3%, 12.5%, 62.5% respectiv 100% din setul de antranament. Atunci cand datele de antrenare sunt intr-un numar mai redus, observam ca am obtinut o performanta mai mare cu Bayes Naiv: 0.755, comaprat cu Regresia Logisitca: 0.750. Regresia Logistica a ajuns destul de rapid din urma insa algoritmul Bayes Naiv, ca in final performanta acesteia sa fie mai buna: 0.98 vs 0.91.

4.3 Concluzii Capitol

Dupa cum am aratat mai sus, atunci cand ne confruntam cu o problema de clasificare exista doua moduri de abordare: estimarea directa a parametrilor probabilitatii $P(Y \mid X)$ - ceea ce numim model discriminativ - sau estimare paramatrilor pentru $P(y)$ si $P(X \mid Y)$, acesta din urma purtand numele de model generativ.

De asemenea, am mai arătat că făcând presupunera că $P(X \mid Y)$ urmează o distribuţie Gaussiană am ajuns la formă probabilităţii $P(Y \mid X)$ care este folosită la Regresia Logisica. În contiuare, pentru a prezenţa câteva avantaje şi dezavantaje pentru modelul generativ, respectiv discriminativ vom folosi că exemple algoritmul generativ Bayes Naiv Gaussian (GNB), respectiv algoritmul discriminativ Regresia Logistică, ambele prezentate mai sus în cadrul acestei lucrări.

Atunci când presupunerile făcute de algoritmul Bayes Naiv Gaussian nu sunt îndeplinite, Regresia Logisitica şi GNB învaţă funcţii diferite de clasificare. În acest caz eroarea asimptotică a Regresiei Logistice este mai mică decât în cazul algoritmului Bayes Naiv Gaussian. Pe de altă parte, GNB şi Regresia Logistică converg la eroarea lor asimptotică la rate diferite. După cum a fost demonstrat într-o lucrare publicată de Ng Jordan (2002), pe măsură ce numărul de instanţe de antrenare n creşte, iniţial GNB va da rezultate mai bune dar Regresia Logistică va ajunge la un moment dat să aibă acurateţe mai mare.

Concluzionand:

Modelul Generativ

Avantaje:

- Ne oferă o viziune asupra distribuţiei datelor
- Converge la eroarea asimptoptica setul de rapid

Dezavantaje:

- Are nevoie de mulţi parametri

Modelul Discriminativ

Avantaje:

- Nu depinde de presupunerea de independenţa condiţională a datelor
- Uşor de modelat

Dezavantaje:

- Nu este potrivit atunci când dorim să ştim cum au fost generate datele

5 Prezicerea Cancerului la san

5.1 Introducere

În România mortalitatea cauzată de cancerul la sân este de 36%, ceea ce este mult peste media mondială. Conform unei statistici, 90% din cazurile noi de cancer sunt diagnosticate foarte târziu, acestea fiind deja în stadiile ÎI, III sau IV. Deşi o parte din vina o poartă pacienţii pentru că merg prea târziu la medic, o altă cauza de alarmare este numărul redus de resurse pe care România îl aloca pentru această problemă. Conform unui doctor oncolog din România, dr Dan Jinga: ”În momentul de fată suntem doar 360 de oncologi în România. Aceşti 360 de oncologi impreună cu 140 de hematologi, adică 500 de oameni sunt implicaţi în administrarea tratamentelor pentru 105.000 de pacienţi în fiecare an. Fiecărui medic îi revine în România în medie 450 de proceduri terapeutice pe lună, enorm. Din punctul de vedere al oncologilor, abia avem timp să tratăm pacienţii bolnavi.” În aceste condiţii, un sistem automat ar fi foarte benefic pentru ajutarea medicilor în diagnosticarea acestei boli. În plus, adunând destule date de antrenare şi obţinând o acurateţe bună, acest sistem va elimină nevoia că pacienţii să mai treacă prin numeroase ţeste medicale, care expun pacienţii la dureri şi radiaţii.

De-a lungul timpului, au fost propuşi mai mulţi candidaţi pentru biomarkerii cancerului la sân. În 1994, a fost desfăşurat un studiu ”Machine Learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates”. În acest studiu au fost analizate 30 de atribute extrase din imaginile scanate. Pe aceste date a fost aplicat un algoritm numit ”Multi-surface Method” care încearcă să plaseze o serie de hiperplanuri în spaţiul dimensiunilor de date, încercând să minimizese numaul de hiperplanuri. Această tehnică a reuşit să obţină o acurateţe între 95% şi 98%. În 2012, au fost propuşi alţi 12 biomarkeri(Osteopontin, Haptoglobina, CA15-3, Antigenul Carcinoembrionic, Antingenul Cancer 123, Prolactină, Antingenul Cancer 19-9, *alpha*-fetoproteina, leptina şi factorul migrării) însă nu s-a reuşit colectarea unor observaţii semnificative pentru a face o prezicere bună. O cantitate mare de muncă a fost depusă în analiză seturilor de date făcute publice în baza de date ”UCI Machine Learning Repository”: ”Wisonsin Breast Cancer Data Set”, ”Wisconsin Diagnosis Breast Cancer” şi ”Wisonsin Prognosis Breast Cancer”. (Bibliografie). Aceste seturi de date au fost colectate de la pacienţi în anul 1995. Cele mai multe lucrări

au experimentat pe aceste seturi de date cu diferiţi algoritmi de învăţare automată, printre care cei mai folosiţi au fost: arbori de decizie, reţele neuronale şi Maşini cu vector suport.

În continuare, vom aplica algoritmii descrişi în această lucrare pe un set de date colectat în 2018 şi vom compara modul în care cele două regimuri de abordare, discriminativ respectiv generativ, influenţează modul în care alforitmul face o predicţie.

5.2 Date Utilizate

Setul de date a fost obţinut din baza de date "UCI Machine Learning Repository" şi au fost colectate de la un spital din Coimbra. În total sunt 116 de observaţii, cu 64 de cazuri pacienţi bolnavi şi 52 sănătoşi. Fiecare rând conţine 9 atribute diferite şi o clasificare: 1 sau 2 reprezentând un pacient sănătos respectiv bolnav. Cele 9 atribute reprezintă date care pot fi colectate din simple analize de sânge şi anume: Vârstă, Index de greutate, nivel glucoză, nivel insulină, indice HOMA, nivel leptina, adiponectina, rezistina şi nivelul proteinei MCP-1.

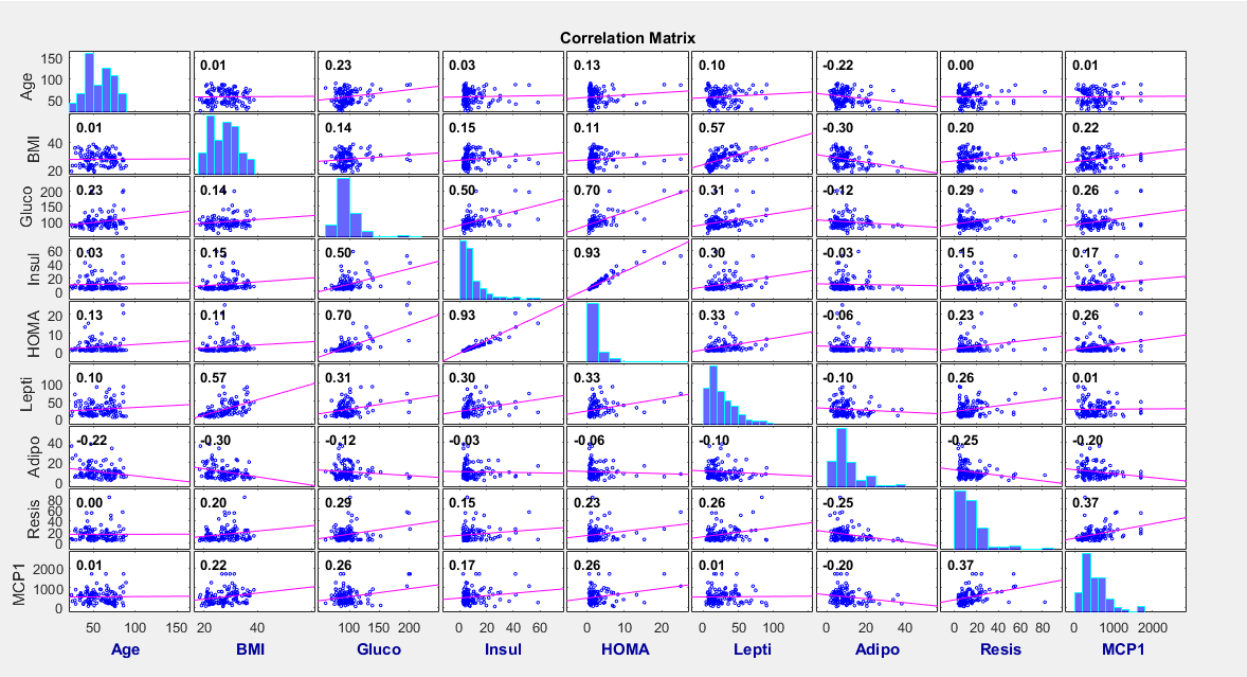
Întrucât setul de date are o dimensiune destul de mică, pentru a testa acurateţea algoritmilor vom aplica cross validarea cu părţile [3, 5, 7, 10]. Astfel, pentru fiecare număr n din mulţimea menţionată, vom împarţi setul de date în n părţi egale, vom testa algoritmul pe una din părţi şi cu restul vom face antrenarea.

5.2.1 Alegerea Atributelor

În dorinţă de a obţine o acurateţe cât mai mare, primul pas va fi să alegem cele mai semnfiicative atribute din cele 9. Vom încerca să obţinem un număr de 3 sau 4 atribute, ceea ce este mult mai potrivit pentru un set de date aşa mic. Pentru a alege atributele, vom aplică mai multe metode iar în final vom alege atributele care sunt clasificate în top 4 trăsături cel mai des.

1) Matricea de Corelaţie Mai întâi am calculat matricea de corelaţie dintre variabile, disponibilă în figura de mai jos. Un index de corelaţie mai mare indică faptul că două atribute sunt destul de asemănătoare şi deci alegerea unui singur atribut din cele două este de ajuns. După cum putem

Figura 8: Matricea de Corelaţie



observa în figura de mai sus, indicele HOMA şi nivelul de Insulină ar putea fi grupate împreună, acestea având un nivel de corelaţie de 0.93.

2) În continuare vom aplica o serie de algoritmi din libraria ”Feature Selection Library”, distribuita open-source in cadrul platformei Matlab.

- Infinite Latent Feature Slection: acest algoritm construieşte un graf având ca noduri atributele iar muchiile modelează relaţia dintre trăsături, mai exact verosimilitatea ca nodurile x şi y să fie candidaţi buni pentru selecţie. În urma aplicării acestui algoritm am obtiunut următoarea ordine a trăsăturilor în funcţie de importanţa lor: [nivel leptina, nivel glucoză, Vârstă, nivel insulină, indice HOMA, adiponectina, MCP-1, Index de greutate, rezistina]

- Recursive Feature Elimation: acest algoritm începe cu setul iniţial de atribute şi apoi elimină pe rând câte un atribut care este mai puţin important până se ajunge la numărul dorit de atribute. În urma aplicării acestui algoritm am obţinut: [Vârstă, Index de greutate, nivel glucoză, nivel insulină, indice HOMA, nivel leptina, adiponectina, rezistina, MCP-1]

- Scorul Laplacin: acest algoritm se bazează pe observaţia că în problemele reale de clasificare, observaţiile din aceaşi clasa sunt în general apropiate una de alta. În urma aplicării acestui algoritm am obţinut: [MCP-1, nivel glucoză, rezistina, nivel leptina, Vârstă, Index de greutate, nivel insulină, adiponectina, indice HOMA]

- Scorul Fisher: în acest algoritm, atributele sunt ordanate după un scor calculat pt fiecare atribut în$_p arte$:

$$scor(s_i) = \frac{\sum n_j(\mu_{ij} - \mu_i)^2}{\sum n_j * \sigma_{ij}^2} \tag{49}$$

  unde μ_{ij} si σ_{ij} sunt media şi varianţa atributului în$_c lasa$ instanţe în j, n_j este numărul de clasa i În j iar $mu_u rma$ este media observaţiei glucoză. insulină Vârstă acestui algoritm am obţinut: [nivel , indice HOMA, nivel , Index de greutate, rezistina, , MCP-1, nivel leptina, adiponectina]

După ce am rulat aceşti algoritmi am observat că atributele Vârstă, index de Greutate şi index Glucoză apăr cel mai frecvent în top 4 atribute.

5.3 Metode şi Rezultate

În continuare vom folosi algoritmii decrisi în această lucrare pentru a găsi un model care să descrie setul de date. Astfel, vom rula întâi un algoritm generativ: Bayes Naiv, împreună cu variatia lui Bayes Naiv Gaussian, iar apoi 2 algoritmi discriminativi: Regresia Logistică (folosind metodă gradientului ascendend şi metoda celor mai mici pătrate) şi Maşinile cu Vector Suport. Pentru a testa modelul, vom folosi cross-validarea pe 4 niveluri: 3, 5, 7, 10. Astfel, pentru fiecare nivel n, vom împarţi setul de date în n părţi egale şi vom testa pe o singură parte, după ce am antrenat pe restul din setul de date. Fiecare algoritm a fost rulat de 100 de ori, de fiecare dată alegând observaţii random pentru datele de antrenare respectiv testare. În final, am calculat media preciziilor obţinute.

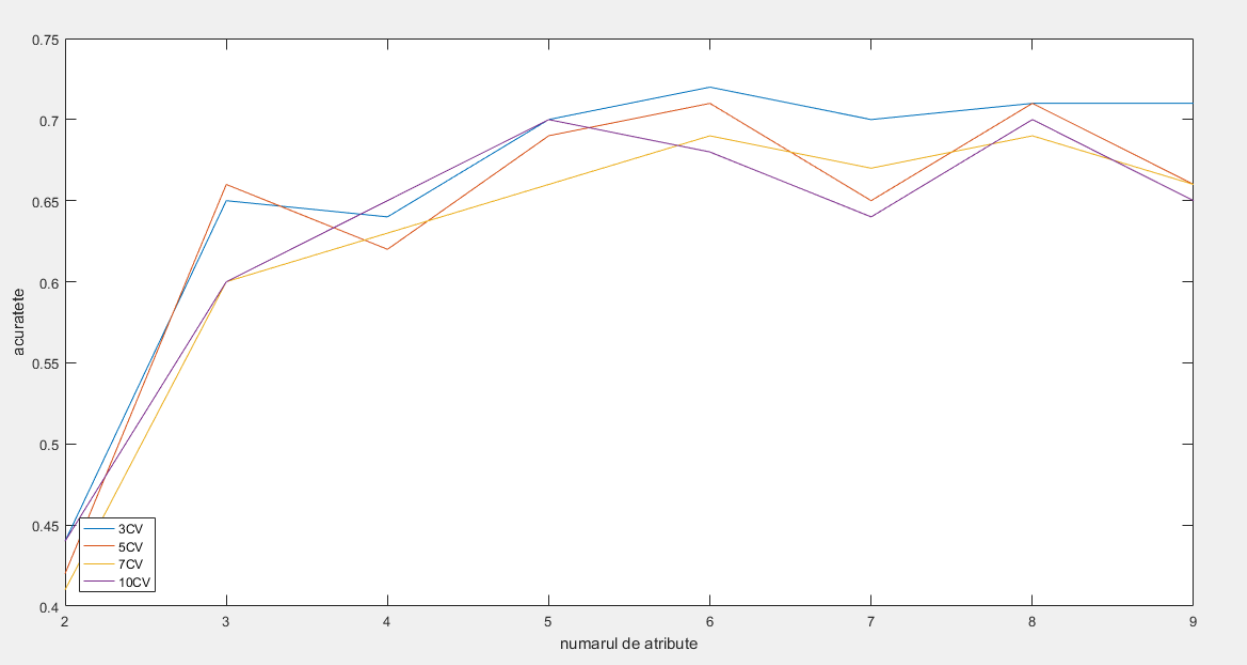
În tabelul de mai jos am pus rezultatele obţinute înainte de a aplica un algoritm de alegere a trasturilor

	3-fold CV	5-fold CV	7-fold CV	10-fold CV
NB	0.51	0.57	0.57	0.57
GNB	0.63	0.62	0.63	0.62
LR IRLS	0.71	0.7	0.74	0.73
LR Gr. des.	0.7	0.61	0.55	0.56
LR Kernelizata	0.71	0.66	0.66	0.65
SVM	0.76	0.75	0.7	0.7

În urma aplicării algoritmilor de selecţie am rulat din nou algoritmii de mai sus, de data aceasta folosind doar atributele: Vârstă, Index Greutate, nivel glucoză şi rezistina.

	3-fold CV	5-fold CV	7-fold CV	10-fold CV
NB	0.66	0.63	0.63	0.63
GNB	0.66	0.7	0.68	0.67
LR IRLS	0.75	0.76	0.75	0.72
LR Gr. des.	0.55	0.56	0.56	0.57
LR Kernelizata	0.65	0.61	0.64	0.64
SVM	0.82	0.78	0.8	0.82

Figura 9: Aplicare RFE pentru LRK



5.4 Concluzii

Observăm că algoritmul Bayes Naiv a dat rezultate mai bune după aplicarea algoritmului de selecţie al atributelor. Acest lucru este din cauză că, după cum am observat în matricea de corelaţie, avem variabile corelate între ele, ceea ce nu este bine pentru algoritmul Bayes Naiv care face presupunerea că datele sunt independente. De asemenea algoritmul Bayes Naiv Gaussian a dat rezultate mai bune decât Bayes Naiv, deci sunt şanse că presupunerea că datele să urmeze o distribuţie Gaussiană să fie adevărată.

Ceea ce este interesant este faptul că obţinem rezultate mai bune folosind clasificatorii discriminativi, deşi setul de date este relativ mic. De asemenea, observăm că obţinem rezultate mai bune folosind Regresia Logistică cu metoda iterativă a celor mai mici pătrate (IRLS) decât cu metodata gradientului ascendent. Acest lucru poate fi din cauza că metodă IRLS deşi este mai complexă şi are nevoie de mai multă putere computaţională, este mai rapidă. Astfel, după 100 de iteraţii regresia logistică cu metooda IRLS a ajuns la o acurateţe mai bună decât regresia logistică folosind gradientul ascendent după 5000 de iteraţii. Pentru a obţine rezultate mai bune şi cu regresia logistică folosind gradientul ascendent algoritmul ar trebui ruulat de mult mai multe ori.

Deşi Regresia Logistică IRLS a avut rezultate similare cu algoritmul Maşini cu Vector Suport înainte de selecţia atributelor, după selecţie Maşinile cu Vector Suport au crescut semnificativ în acurateţe. Acest lucru ne indică faptul că cele 4 atribute alese de selecţie au fost transpuse într-un spaţiu dimensional în care acestea sunt relativ separabile liniar.

5.5 Direcţii Viitoare

Alegerea unor trăsături semnificative ne-a ajutat să obţinem o acurateţe mai mare, însă acest procent tot nu este suficient de mare pentru a prezice cu succes apariţia cancerului la sân la un pacient. Întrucât setul de date a conţinut doar 116 observaţii, probabil un set de date mai mare va duce la o performanţă mai bună.

6 Bibliografie

1. T. Mitchell, Machine Learning, Chapter 2: Estimating Probabilities: http://www.cs.cmu.edu/~tom/mlbook/Joint_MLE_MAP.pdf

2. T. Mitchell, Machine Learning, Chapter 3:GENERATIVE AND DISCRIMINATIVE CLASSIFIERS: NAIVE BAYES AND LOGISTIC REGRESSION: http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf

3. Andrew Ng. Michael Jordan: On discriminative vs Generative classifiers: A comparison of logistic regression and Naive Bayes

4. Liviu Ciortuz et al: Exercitii de invatare automata

5. Liviu Ciortuz, curs 6: Bayesian Learrning: https://profs.info.uaic.ro/~ciortuz/SLIDES/ml6.pdf

6. Tony Jebara, Discriminative, Generative and Imitative Learning: http://www.cs.columbia.edu/~jebara/papers/jebara4.pdf

7. Julia Aurelie Lassere, Hybrids of Generative and Discriminative Methods for Machine Learning http://mi.eng.cam.ac.uk/~jal62/publis/thesis.pdf

8. Eric Xing, CMU, Lecture 5: https://www.cs.cmu.edu/~epxing/Class/10701-10s/Lecture/lecture5.pdf

9. Jing Hao Xue, Aspects of Generative and Discriminative Classifiers: http://theses.gla.ac.uk/272/1/2008xuephd.pdf

10. Pattern Recognition and Machine Learning - Christopher M. Bishop

11. Pedro Alonso, A comparison between some discriminative and generative classifiers

12. Andrew Ng, Standford CS229 Lecture notes, Part IV: http://cs229.stanford.edu/notes/cs229-notes2.pdf

13. Using Resistin, glucose, age and BMI to predict the presence of breast cance https://bmccancer.biomedcentral.com/articles/10.1186/s12885-017-3877-1

14. EduMedical: https://www.edumedical.ro/prevalenta-in-cazurile-de\newline-cancer-la-san-a-crescut-enorm-in-romania/

15. Masini cu Vector Suport: https://towardsdatascience.com/support-vector\newline-machines-a-brief-overview-37e018ae310f