Detektovanje spam komentara korišćenjem klasifikatora

Marko Mihajlović 14742

April 17, 2017

Danas postoji veliki broj sajtova na internetu na kojima korisnik može postavljati svoje komentare bez ikakve cenzure. Mnogi komentari nisu prikladni i ne želimo ih na sajtu. Ovakve komentare nazivamo spam komentarima. Najčešće spam komentare generišu deca koja nisu svesna generisanja javnog kontenta.

U ovom radu je prikazan jedan od načina za detektovanje spam komentara. U prvom delu rada je opisan klasifikacioni problem i metode za klasifikaciju podataka koje mogu biti korišćene za implementaciju rešenja, nakon toga je opisana logistička regresija i linearna diskriminenta anliza, dok je na kraju na kraju opisana praktična implementacija sistema i prikazano poređenje rezultata različitih klasifajera.

1 Klasifikacija - klasifikacioni problem

Klasifikacija predstavlja vrstu mašinskog učenja, koja je podoblast veštačke intaligencije čiji je cilj konstruisanje algoritama i računarskih sistema koji su sposobni da se adaptiraju na nove situacije i uče na osnovu iskustva. Razvijene su različite tehnike učenja za izvršavanje različitih zadataka. Osnovne tehnike se tiču nadgledanog učenja za diskreciono donošenje odluka, nadgledanog učenja za kontinuirano predviđanje i pojačano učenje za sekvencionalno donošenje odluka, kao i nenadgledano učenje.

Većina praktičnih problema koristi oblik nadgedanog mašinskog učenja. Ovaj model podrazumeva primenu nekog algoritma nad skupom ulaznih X i izlaznih promenljivih Y, trening podaci, za učenje mapiranja $Y = f(X)$. Cilj je proceniti parametre funkcije f tako da se ova funkcija može primeniti za nove ulazne podatke X za koje ne znamo izlaz Y, test podaci. Podela nadgledanog učenja:

- **Klasifikacija**: Problem identifikovanja kategorije klase novog posmatranja.
- Regresija: Problem predikcije kvantitivne vrednosti.

Nenadgledano učenje za ulazne podatke X modelira strukture podataka ili distribuciju podataka bez povratnih informacija Y. Cilj je uočavanje zajedničkih svojstava podataka. Ovaj oblik učenja možemo svrstati:

- Klasterizacija - metod za analizu grupisanja čiji je cilj particionisanje ulznih podataka na k klastera.
- Asocijacija - metod za generaisanje pravila koja opisuju podatke.

Javlja se još jedan oblik mašinskog učenja, polu-nadgledano učenje. Labele su dodeljene manjim brojem ulaznih podataka. Razlog može biti cena ručnog procesiranja informacija.

Problem sa kojim se mi srećemo je upravo kvalitativne prirode, gde je potrbno predvideti da li je komentar spam ili ne. Matematički ovu vrednost možemo predstaviti kao binarnu vrednost (spam - 0, nije spam - 1).

Za klasifikaciju podataka se mogu koristiti klasifikatori kao što su logistička regresija, linearna diskriminantna analiza, k najbližih komšija, random forest, stabla, support vector classifiers i drugi. Problem klasifikacije je složen i ne postoji univerzalan klasifikator koji će raditi najbolje u svim situacijama.

2 Logsistička regresija

Algoritam od kog potiče logistička regresija se naziva linearna regresija, koja predstavlja algoritam nadgledanog mašinskog učenja koji je našao primenu u regresiji, njegova modifikacija nalazi primenu u rešavanju klasifikacionog problema.

Linearna regresija je predstavljena linearnom funkcijom odučivanja oblika:

$$y \;=\; \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n + \epsilon \tag{1}$$

gde ϵ predstavlja grešku koja ima normalnu raspodelu i predstavlja odstupanje dobijene vrednosti u odnosu na izlaz y.

Jasno je da nam ovakva reprezentacija ne odgovara za klasifikovanje diskretnog izlaza. Umesto direktnog predviđanja klase Y, logistička regresija modelira verovatnoću $p\left(X\right)$ da X pripada specifičnoj kategoriji. Postavljanjem odgovarajuće granice ($threshold$) možemo izvršiti diskretizaciju izlaza:

$$Y = \begin{cases} 0, & \text{if } p\left(x\right) < threshold \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

Pitanje kojim želimo da se bavimo, je kako modelirati vezu između $p\left(X\right) = \theta_0 + \theta_1 X$ i X? Zbog jednostavnosti problema smatraćemo da izlaz ima binarnu vrednost, nula ili jedan.

Najjedenostavnije rešenje je korstiti linearnu regresiju za predstavljanje verovatnoće:

$$p\left(X\right) = \theta_0 + \theta_1 X \tag{3}$$

Problem kod ovakvog predstavljanja je dobijanje vrednosti koja je negativna ili veća od jedan za specifične ulaze, verovatnoća mora biti u intervalu $[0, 1]$. Kako bismo izbegli ovaj problem, modeliraćemo verovatnoću $p\left(X\right)$ koristeći funkciju koja daje izlaz između 0 i 1 za sve vrednosti X. Mnoge funkcije zadovoljavaju ovu osobinu, u logsitičkoj regresiji koristimo logističku funkciju (sigmoid):

$$\sigma\left(x\right) = \frac{e^t}{1 + e^t} \tag{4}$$

zamenom u funkciji (3) dobijamo:

$$p\left(x\right) = \frac{e^{\theta_0 + \theta_1 X}}{1 + e^{\theta_0 + \theta_1 X}} \tag{5}$$

Zbog matematičke pogodnosti verovatnoću možemo predstaviti preko šanse ($odds$):

$$odds = \frac{p\left(x\right)}{1 - p\left(x\right)} \tag{6}$$

primenom ove formule nad (5) dobijamo:

$$\frac{p\left(x\right)}{1 - p\left(x\right)} = e^{\theta_0 + \theta_1 X} \tag{7}$$

zatim logaritmovanjem:

$$\log \frac{p\left(x\right)}{1 - p\left(x\right)} = \theta_0 + \theta_1 X \tag{8}$$

dobijamo funkciju koja se naziva $logit$ i koja je linearna po X.

Za procenu parametara θ_0 i θ_1 koristimo likelihood funkciju:

$$l\left(\theta_0, \theta_1\right) = \prod_{i: y_i = 1} p\left(x_i\right) \prod_{\hat{i}: \hat{y_i} = 0} \left(1 - p\left(\hat{x_i}\right)\right) \tag{9}$$

θ_0 i θ_1 se biraju tako da maksimizuju (9).

Generalizacijom ovog modela dobijamo multiple logistic regression:

$$\log \frac{p\left(x\right)}{1 - p\left(x\right)} = \theta_0 + \theta_1 X_1 + \cdots + \theta_p X_p \tag{10}$$

gde je $X = (X_1, X_2, \cdots, X_p)$, a p redni proj prediktora. Ova jednačina se može zapisati kao:

$$p(x) = \frac{e^{\theta_0 + \theta_1 X_1 + \cdots + \theta_p X_p}}{1 + e^{\theta_0 + \theta_1 X_1 + \cdots + \theta_p X_p}} \tag{11}$$

isto kao i u prethodnom primeru, procena $\theta_0, \theta_1, \cdots \theta_p$ se vrši korišćenjem maximum likelihood funkcije.

Nedostatak logistička regresije se javlja kod klasifikovanja odziva koji može da ima više od dve klase. Sa ovim modelom se mora pribegavati višestrukim korišćenjem binarne klasifikacije (strategije one versus all, one versu one). Metod koji je obrađen u nastavku, linearna diskriminantna analiza, je pogodniji za multiple-class klasifikaciju.

3 Linearna diskriminentna analiza - LDA

Osnovni nedostatak prethodnog modela je nepogodnost predikcije nebinarnog izlaza. Ovo nije slučaj sa Linearnom diskriminentnom analizom. Pored ove razlike, logistička regresija je vrlo nestabilna sa dobro odvojenim klasama. LDA je stabilniji klasifajer i kada je broj prediktora X mali i ima približno normalnu distribuciju.

Za razliku od logističke regresije koja direktno modelira verovatnoću $Pr\left(Y=k|X=x\right)$ koristeći logističku funkciju (11), LDA ima manje direktan pristup. Naime, procena verovatnoće se vrši uz modeliranje distribucije prediktora X odvojeno za svaku od rezltujućih klasa Y, i nakon toga, koristi Bajesove teoreme za prebacivanje ovih procena u rezultujuću verovatnoću $Pr\left(Y=k|X=x\right)$. Kada su distribucije X normalne onda je model sličan logističkoj regresiji.

3.1 Bajesova teorema za klasifikaciju

Pretpostavimo da želimo da klasifikujemo podatke u k klasa, $k \geq 2$. Sada kvantitivni izlaz Y može imati k različitih vrednosti. Uzmimo da π_k predstavlja verovatnoću da posmatranje x_i pripada klasi k, i da $f_k\left(X\right) \equiv Pr\left(X=x|Y=k\right)$ označava funkciju gustine od X da jedno posmatranje pripada klasi k. Prema Bajesovoj teorimi:

$$p_k\left(x\right) = Pr(X=x|Y=k) = \frac{\pi_k f_k\left(x\right)}{\sum_{l=1}^{K} \pi_l f_l\left(x\right)} \tag{12}$$

π_k predstavlja frakciju trening podataka koji pripadaju trening skupu, dok je procena $f_k\left(X\right)$ malo zahtevnija. Procenom ove funkcije dobijamo verovatnoću za klasifikovanje podataka.

3.2 LDA za jedan prediktor $p = 1$

Pretpostavimo da imamo samo jedan prediktor. Želimo da procenimo $f_k\left(x\right)$ kako bismo uz pomoć (12) odredili $p_k\left(x\right)$ i klasifikovali podatak x određenoj klasi za koju je $p_k\left(x\right)$ najveći. Da bi procenili $f_k\left(x\right)$, napravićemo par pretpostavki.

Smatrajmo da $f_k\left(x\right)$ ima Gausovu raspodelu. U jednodimenzionalnim uslovima normalna gustina ima oblik:

$$f_k\left(x\right) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2\sigma_k^2}\left(x-\mu_k\right)^2} \tag{13}$$

gde je μ_k srednja vrednost, a σ_k^2 varijansa za k-tu klasu. Zatim smatrajmo da imamo istu varijansu za svaku klasu ($\sigma^2 = \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2$). Iz (12) i (13) dobijamo:

$$p_k\left(x\right) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}\left(x-\mu_k\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}\left(x-\mu_l\right)^2}} \tag{14}$$

Bajesov klasifajer dodeljuje jednom podatku $X = x$ klasu za koji je (14) najveći. Logaritmovanjem (14) i sređivanjem izraza dobijamo ekvivalentan zapis

$$\delta_k\left(x\right) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log \pi_k \tag{15}$$

za koji je $\delta_k\left(x\right)$ najveći.

Sada je potrebno proceniti π_k, μ_k i σ^2 kako bi izračunali $\delta_k(x)$:

$$\hat{\mu} = \frac{1}{n_k} \sum_{i:y_i=k} x_i \tag{16}$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu_k})^2 \tag{17}$$

$$\hat{\pi_k} = \frac{n_k}{n} \tag{18}$$

n je ukupan broj trening podataka, n_k broj trening podataka koji pripadaju klasi k, μ_k srednja vrednost svih trening podataka iz klase k, a σ težinska srednja vrednost varijanse za svaku od K klasa.

Integrisanjem (16), (17) i (18) u (15) dobijamo procenenju vrednost $\hat{\delta}_k(x)$ na osnovu čije maksimalne vrednosti dodeljujemo klasu ulaznom podataku $X = x$.

$$\hat{\delta}_k(x) = x\frac{\hat{\mu_k}}{\hat{\sigma}^2} - \frac{\hat{\mu_k}^2}{2\hat{\sigma}^2} + \log\hat{\pi_k} \tag{19}$$

Naziv linearan u LDA upravo potiče iz činjenice da je diskrimitivna funkcija $\hat{\delta_k}(x)$ u (19) linearna po x.

3.3 LDA za veći broj prediktora $p > 1$

Za veći broj prediktora $X = (X_1, X_2, \cdots, X_p)$ primenićemo multivariacionu normalnu distribiciju sa specifičnim vektorom srednjih vrednosti i zajedničkom matricom kovarjansi.

P-dimenzionalni vektor X ima multivariacionu normalnu distribiciju i obeležavamo sa $X \sim N(\mu, \sum)$, srednja vrednost od X je $E(X) = \mu$ i p x p matrica kovarjansi $Cov(X) = \sum$. Za ove parametre dobijamo multivariacionu Gausovu funkciju gustine:

$$f(x) = \frac{1}{(2\pi)^{\pi/2}|\sum|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu)} \tag{20}$$

sada, integrisanjem ove funkcije gustine u (12) dobijamo funkciju čijem ulazu x vrednost je klasu za koju je vrednost:

$$\delta_k(x) = x^T \sum^{-1} \mu_k - \frac{1}{2}\mu_k^T \sum^{-1} \mu_k + \log\pi_k \tag{21}$$

najveća.

3.4 Kvadratna diskriminentna anliza - QDA

Nedostatak LDA-a je koršćenje iste matrice kovarjansi za svaku klasu. Kod QDA-a svaka klasa ima svoju mtricu kovarjansi. Podatak iz klase k je oblika $X \sim N(\mu_k, \sum_k)$, gde je $\sum_k$ matrica kovarjanse za k-tu klasu. Bajesov klasifajer dodeljuje podatak klasi za koju je

$$\hat{\delta_k}(x) = -\frac{1}{2}(x-\mu_k)^T \sum_k^{-1}(x-\mu_k) + \log\pi_k \tag{22}$$

najveće. Ovaj funkcija je kvadratna, otuda i naziv quadratic discriminant analysis.

Zašto QDA klasifikuje podatke sa većom pouzdanošću u odnosu na LDA, odgovor leži u bias-variance trade-off. Ako je potrebno proceniti p prediktora, onda procena matrice kovarjansi zahteva procenu $p(p+1)/2$ parametara, QDA procenjuje za svaku matricu posebno, $K * p(p+1)/2$. Za veliki broj parametara ovo može biti vremenski veoma zahtevno.

Trade-off: LDA smatra da je matrica kovarjansi ista između svih klasa, što je loše zbog toga što LDA može patiti od visokog bias-a. LDA je bolji od QDA kada je mali broj trening podataka i smanjivanje varijanse je krucijalno. Sa druge strane QDA je preporučljiv kada je obiman trening skup, tako da varijansa klasifajera nije u prvom planu. Sa druge strane, LDA je manje fleksibilan što se ugleda u manjoj varijansi, što zanči da se kod LDA može javiti problem visokog bias-a.

4 Mašine potpornih vektora - SVM

Mašine potpornih vektora (support vecotr machine - SVM) spadaju u grupu neprobalističkih modela nadgledanog obučavanja i predstavlja binarni linearni klasifikator - klasifikuje objekte u dve kategorije. SVM je generalizacija jednostavnog klasifikatora koji se naziva *maximal margin classifier*.

4.1 *Maximal Margin Classifier*

U nastavku ovog poglavlja opisan je proces klasifikacije uz pomoć hiperravni, kao i koncept optimalnog konstruisanja klasifikatora.

4.1.1 Sta je hiperravn?

Hiperravan je ravan koja je u p-dimenzionalnom prostoru $p-1$ dimenzionalni podprostor. Na primer, u dvodimenzionalnom prostru hiperravan je jednodimenzionalni podporstor, dok je u trodimenzionalnom prostoru hiperravn definisana kao $\theta_0 + \theta_1 X_1 + \theta_2 X_2 = 0$, a p-dimenzionalna hiperravan:

$$\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_p X_p = 0 \tag{23}$$

Hiperravan možemo posmatrati kao ravan koja deli p-dimenzionalni prostor na dva dela. U tom slučaju posmatramo (23) kao nejednačine (24, 25), na osnovu koje se određuje pripadnost klasi.

$$\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_p X_p > 0 \tag{24}$$

$$\theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_p X_p \leq 0 \tag{25}$$

Određivanje klase se jednostavno svodi na ispitivanje znaka leve strane jednačine (23).

4.1.2 Klasifikovanje korišćenjem hiperravni

Kako bi izvršili klasifikaciju, najpre nam je potrebna matrica podataka X, pri čemu se ona sastoji od n trening podataka u p-dimenzionalnom prostoru,

$$x_1 = \begin{pmatrix} x_{11} \\ x_{22} \\ \vdots \\ x_{1p} \end{pmatrix}, \cdots, \; x_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{np} \end{pmatrix} \tag{26}$$

i ova posmatranja pripadaju jednoj od dve klase - $y_1, \; y_2, \; y_n \in \{-1, \; 1\}$, gde -1 predstavlja jednu, a 1 drugu klasu. Pored toga potreban nam je i skup podataka za testiranje $x^* = (x_1^*, \; x_2^*, \; \cdots, x_n^*)^T$. Naš cilj je razviti klasifikator na osnovu trening podataka tako da tačno klasifikuje test podatke.

Sada (24) i (25) možemo zapisati kao:

$$\theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_p X_{ip} > 0 \quad if \;\; y_i = 1, \tag{27}$$

i

$$\theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_p X_{ip} \leq 0 \quad if \;\; y_i = -1, \tag{28}$$

Ekvivalentno, hiperravan ima osobinu

$$y_i \left(\theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_p X_{ip}\right) > 0, \quad \forall i = 1, \; \cdots, \; n. \tag{29}$$

Sada možemo vršiti klasifikovanje test podataka na osnovu znaka funkcije

$$f(x^*) = \theta_0 + \theta_1 x_1^* + \theta_2 x_2^* + \cdots + \theta_p x_p^*.$$

Ako je $f(x^*)$ pozitivno onda dodeljujemo podatku klasu 1, a ako ne onda -1. Pored ove osobine, na osnovu apsolutne vrednosti $f(x^*)$ možemo odrediti i koliko je trenutno posmatranje x^* daleko od hiperravni. Mala vrednost $f(x^*)$ ukazuje da je velika verovatnoća pogrešne klasifikacije, dok velika vrednost zanči da je posmatranje klasifikovano sa velikom pouzdanošću.

4.1.3 Klasifikator najveće margine - Maximal Margin Classifier

Iz prethodnog izlaganja se može uočiti prvi problem idealnog postavljanja hiperravni. U većini slučajeva, podaci mogu biti podeljeni na bezbroj načina, međutim, način odabira pozicije hiperravni koji se je pokazao kao najefektivniji se svodi na slučaj kada je margina najveća.

Izračunavanje normalne distance svakog trening podataka nam može omogućiti idealno određivanje pozicije ravni, pri čemu se najmanja takva distanca naziva marginom. Hiperravan na osnovu koje se vrši podela je određena tako da je margina najveća, tj. hiperravan za koju je najudaljenija minimalna distanca nad trening podacima. Nakon toga, možemo klasifikovati podatke na osnovu koje strane se nalaze test podaci u odnosu na hiperravan. Ovakav način klasifikacije predstavlja *maximal margin classifier*.

Maximal margin classifier klasifikuje test podatke x^* na osnovu znaka funkcije $f(x^*) = \theta_0 + \theta_1 x_1^* + \theta_2 x_2^* + \cdots + \theta_p x_p^*$, gde su $\theta_0,\ \theta_1,\ \cdots,\ \theta_p$ koeficijenti *maximal margin* hiperravni.

U ovom slučaju, trening podaci koji razdvajaju određuju širinu margine se nazivaju potporni vektori - *support vectors*. Može se uočiti da samo potporni vektori određuju položaj hiperravni što nam daje brojne pogodnosti.

4.1.4 Konstrukcija klasifikatora

Proces konstruisanja klasifikatora za n trening podataka $x_1, x_2, \cdots, x_n \in R^p$ sa dodeljenim labelama $y_1, y_2, \cdots, y_n \in \{-1, 1\}$ koristi optimizaciju kako bi se odredio položaj hiperravni. Optimizacija se svodi na

$$\max_{\theta_0,\cdots,\theta_p} M \tag{30}$$

$$\sum_{j=1}^{p} \theta_j^2 = 1 \tag{31}$$

$$y_i\left(\theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_p X_{ip}\right) \geq M \quad \forall i = 1, \cdots, n \tag{32}$$

Pri čemu ograničenje (32) garantuje da će svaki podatak biti na ispravnoj strani hiperravni, (31) dodaje ograničenje da je normalna distanca od i-tog podatka do hiperravni definisana sa

$$y_i\left(\theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_p X_{ip}\right).$$

Tako da ograničenja (32) i (31) omogućavanju da se podatak nađe sa korektne strane i na najmanjoj distanci M od hiperravni. M predstavlja marginu hiperravni koju je potrebno maksimizovati podešavanjuće parametre $\theta_0,\ \theta_1,\ \cdots,\ \theta_p$. Transformisanjem navedenih funkcija se može dobiti oblik koji je pogodniji za optimizaciju, problem koji se u literaturi naziva *kvadratnim optimizacionim problemom sa linearnim ograničenjima*.

Još jedan problem određivanja hiperravni se javlja u slučajevima kada nije moguće podeliti prostor. Za takav skup podataka kžemo da nije linearno separabilan, i u ovom slučaju optimizacioni problem (30, 31, 32) nema rešenja.

U nastavku će biti obrađen princip meke margine koji proširuje koncept podele prostora uz pomoć hiperravni.

Generalizacija klasifikatora sa maksimalnom marginom na neseparabilnom skupu podataka se naziva klasifikator potpornih vektora (*support vector classifier*).

4.2 Klasifikator potpornih vektora - support vector classifier

Do sada smo razmatrali slučajeve u kojima se podaci mogu linearno razdvojiti, međutim ovo nije čest slučaj. Da bi postigli veću robusnost i veću preciznost klasifikovanja dozvolićemo da nekoliko podatka bude pogrešno klasifikovano. Zbog ove osobine se klasifikator baziran na potpornim vektorima često naziva *soft margin classifier*.

U ovom slučaju optimizacioni problem se svodi na:

$$\max_{\theta_0,\cdots,\theta_p,\epsilon_1,\cdots,\epsilon_n} M \tag{33}$$

$$\sum_{j=1}^{p} \theta_j^2 = 1 \tag{34}$$

$$y_i\left(\theta_0 + \theta_1 X_{i1} + \theta_2 X_{i2} + \cdots + \theta_p X_{ip}\right) \geq M\left(1 - \epsilon_i\right) \quad \forall i = 1, \cdots, n \tag{35}$$

$$\sum_{j=1}^{n} \epsilon_j^2 \leq C, \ \epsilon_i \geq 0 \tag{36}$$

gde je C pozitivna konstanta za podešavanje margine, M je širina margine, a $\epsilon_1, \cdots, \epsilon_n$ su promenljive koje nam govore gde se nalazi i-ti podatak (37). Isto kao i u prethodnom slučaju, test podatak x^* će biti klasifikovan na osnovu znaka funkcije $f\left(x^*\right) = \theta_0 + \theta_1 x_1^* + \theta_2 x_2^* + \ \cdots \ + \theta_p x_p^*$.

$$i - ti\ podatak\ se\ nalazi\ na \begin{cases} korektnoj\ strani & \text{if } \epsilon_i = 0 \\ nekorektnoj\ strani\ margine & \text{if } \epsilon_i > 0 \\ nekorektnoj\ strani\ hiperravni & \text{if } \epsilon_i > 1 \end{cases} \tag{37}$$

Sa druge strane, parametar C ograničava broj pogrešno klasifikovanih podatak koje možemo tolerisati. Drugim rečima, C kontroliše širinu margine, margina je šira kada je C veće. U praksi se najčešće ovaj parametar bira u fazi ukrštene validacije, i predstavlja parametar za kontrolu *bias-variance tradeoff*-a. Ako je C veliko, onda veliki broj podataka učestvuje u određivanju hiperravni, što znači da imamo malu varijansu i veliki bajas, dok malo C nam govori da imamo mali bajas, ali zato veliku varijansu.

Klasifikator potpornih vektora ima veliku robusnost zahvaljujući odlučivanju o granici odlučivanja samo na osnovu malog broja potpornih vektora. Ova osobina predstavlja glavnu razliku u odnosu na LDA čija granica odlučivanja zavisi od srednje vrednosti svakog podatka, sa druge strane logistička regresija je dosta slična klasifikatoru baziranom na potpornim vektorima.

5 Sistem za detktovanje spam komentara

Za implementaciju ovog sistema korišćen je programski jezik python, pomoćna biblioteka za mašinsko učenje scikit-learn i nltk (Natural Language Toolkit) za obradu reči.

Sistem predstavlja konzonlu aplikaciju koja poredi rezultate 4 algoritma za klasifikaciju nad specifiranim skupom podataka:

- Linearna regresija - LR
- Linearna diskriminentna anliza - LDA
- Kvadratna diskriminentna anliza - QDA
- Klasifikacija bazirana na potpornim mašinama - support vector classifier (SVC)

Performanse izvršavanja programa nisu uzete u razmatranje zbog toga što se treniranje klasifikatora vrši samo jednom. Serijalizacijom iztreniranog klasifikatora se može vrišiti predviđanje ostalih ishoda.

5.1 Opis biblioteka

5.1.1 Scikit-learn

Scikit-learn je inicijalno razvijen od strane David Cournapeau na *Google summer of code project* 2007. godine. I danas je projekat sponzorisan od strane: *INRIA*, *Google*, *Tinyclues* i *the Python Software Foundation*.

Scikit-learn je python biblioteka koja nudi veliki algoritama nadgledanog i nenadgledanog mašinskog učenja. Ova biblioteka je izdata pod BSD licencom i koristi sledeće pomoćne biblioteke: *NumPy*, *SciPy*, *Matplotlib*, *IPython*, *Sympy*, *Pandas*.

5.1.2 Nltk

Nltk je najpopularnija pythonova open source biblioteka za kreiranje programa koji uključuju interakciju hljudskog govora. Inicijalno nudi veliki broj korpus podataka i ostalih leksikografskih resursa, kao što je *WordNet*, i biblioteke za procesiranje tekstualnih podataka (classification, tokenization, stemming, tagging, parsing, semantic reasoning).

Iz ove biblioteke koristio sam *WordNetLemmatizer*, *word_tokenize*, *stopwords* i *PorterStemmer*.

5.2 Preprocesiranje podataka

Skup podataka koji je korišćen je organizovan u dva fajla, skup dobrih i loših komentara. Primer dobrog komentara:

> You have improved greatly in the past years. This is probably the best improvement from a artist I have ever seen. I love all the colors you have used and how you used them in the before and after. I just love the improvement greatly. I can't find any other words to describe the improvement other then beautiful, creative, cute, and down right awesome. This is better then I could do currently. There is just so much creativity and beautifulness in the improvement I can't keep but repeat myself over and over again.

Primer spam komentara:

> i do say its beautiful mothalicka. you should make a commision of this for me. its possibly the most hawtest eyes in the hole world. and i've never seen such beautiful hurr. (besides mines of course) the legs and arms are such beaituful. and do not get me started with that peerrfecct face. the smile is right on key. and dem eyebrows are better den mah watercolors. eye lashes are right on the face,so thats good. but tell dis gurl to get some clothes. gawd. anyways good job gurl. bravo. love it. omg. so hawt. 10/10 watercolors im old greggggggggg.

Kako bismo postigli pouzdanost predviđanjanja potrebno je svaki komentar prevesti u niz tokena reči. Pri čemu je za svaku reč potrebno izvršiti normalizaciju, normalizacija uključuje navedene korake u nastavku.

Prvi korak u procesiranju podataka je kreiranja liste dobrih i loših komentara. Kako bismo postigli veću pouzdanost predviđanjanja potrebno je svaki komentar **tokenizovati** na reči koje je potrebno normalizovati.

Osnovna normalizacija reči obuhvata konvertovanje svih velikih slova reči u **mala slova**. Na ovaj način ćemo tretirati The i the isto. Ovo predstavlja osnovnu normalizaciju, nažalost u našem skupu podataka i dalje imamo puno sličnih termina koje je potrebno eliminisati.

Sledeći nivo narmalizacije podrazumeva uklanjanje prefiksa i sufiksa reči, proces koji je poznatiji kao **stemming**. Tri najpozantija algoritma koja se koriste danas su: Porter, Snowball(Porter2), i Lancaster (Paice-Husk). Porter Lancaster je najagresivniji, dok je Porterov algoritam blaži u svođenju reči na osnovni oblik, Porter2 predstavlja optimizovaniu varjantu prethodnog algoritma. U ovom sistemu je korišćen Porterov algoritam zbog prethodnih karakteristika. Primena stimera za reč $lying$ je lie.

Kada smo izvršili otklanjanje prefiksa i sufiksa reči, potrebno je svesti dobije reč na osnovni oblik ($lemma$), ovaj proces je poznat kao **lemmatization**. Primer lemmatizera za reč $women$ je $woman$.

Pored ovih načina normalizacije postoje i dodatne koje nisu uključene u ovaj sistem, recimo identifikovanje nestandardnih reči (identifikovanje brojeva, datuma, skraćenica). Na primer, svaki broj bi mogao biti sveden na isti token, takođe i svaki akronim. Na ovaj način bi vokabular ostao manji što bi poboljšalo preciznost klasifikatora.

5.3 Treniranje klasifikatora

Kako bismo trenirali kalasifajer potrebno je uočiti karakteristike podataka koje utiču na ishod klasifikacije.

Neke od mera primenjenih nad dataset-om u ovom sistemu su:

- Broj karaktera
- Broj jedinstvenih karaktera
- Odnos broja jedinstvenih karaktera i ukupnog broja karaktera
- Broj reči
- Da li je učestalost reči najčešće pojavljivane reči veća od 50%

Nakon formiranja matrice sa pet kolana, potrebno je podeliti skup podataka na skup podataka za treniranje i testiranje.

Za podelu podataka je korišćenja urkštena validacija (*cross-validation*), koja obezbeđuje reprezentativnost uzoraka za testiranje. Podela je izvršena u odnosu 8:2.

5.4 Rezultati

Učenje klasifajera je izvršeno nad 912 komentara, dok je testiranje izvršeno nad 228 podataka. Statistika dobijenih rezultata je prikazana u tabeli [Table1].

Klasifikator	broj uspešnih testova	preciznost
Linearna regresija	211	0.9254
Linearna diskriminentna anliza	212	0.9298
Kvadratna diskriminentna anliza	201	0.8816
Support vector classifier	211	0.9254

Table 1: Poređenje rezultata nad 228 testova

Možemo zaključiti da je najbolje rezultate dala linearna diskriminentna analiza. Razlike između LR, LDA i SVC je veoma mala, dok sa druge strane QDA je zbog veće fleksibilnosti imala više promašaja.

References

[1] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. 2010.

[2] Sandra Stojanović. *Logistička regresija i njena primena u klasifikaciji i pretraživanju informacija*. Univerzitet u Nišu, Elektronski fakultet 2016.

[3] Marko Stevanović. *Primena naivne Bayes-ove i SVM klasifikacije teksta u sistemima za pretražiavanje informacija*. Univerzitet u Nišu, Elektronski fakultet 2014.

[4] Sci-kit learn
`http://scikit-learn.org/`

[5] NLTK
`http://www.nltk.org/`