Dokumentacija za predmenti projekat iz predmeta Softverski algoritmi u sistemima automatskog upravljanja

Dušan Grković RA56/2021
31/05/2024

1. Uvod

1.1 Svrha dokumentacije

Objašnjavanje koda i principe rada koje sam koristio pri istraživačkon projektu.

1.2 Kratko objašnjenje

Svrha projekta je da predvidim na osnovu određenih parametara da li određena osoba ima dijabetes, koristeći anačitičke metode obrade podataka i različite modele mašinskog učenja koje sam podesio, uspeo sam da imam uspešnost predviđanja od oko 90% i proširenje ovakvog modela može da bude beneficijalno za rasterećenje zdravstvenog sistema posebno u okruženjima sa ograničenim zdravstvenim resursima.

2. Opis funkcionalnosti

2.1 Meni

Pomoću menija možemo da palimo i gasimo određene funkcionalnosti projekta bez toga da ulazimo u kod i modifikujemo ga. Omogućava nam da brzo obradimo nov set podataka i prikažemo relevantne informacije.

2.2 Podešavanje parametara

Budući da podešavanje parametra zahteva dosta vremena i resursa, možemo da odaberemo da li pri pokretanju hoćemo da ga koristimo ili da učitamo već izračunate najoptimalnije parametre. Podešavao sam paramtre u ensamble modelima, a nakon što sam zaključio da prosti modeli ne mogu da se dotreniraju kao ensamble modeli, nisam trošio dodatne resurse na podešavanje hiperparametara.

2.3 Eksplorativna analiza

Metode koje koristimo tokom eksplorativne analize:

- Da li postoje prazna polja.
- Broj duplikata.
- Histogrami (da dobijemo osećaj koliko podataka ima i da vidimo da li imamo i koliko anomalija).
- Box plot, omogućava nam da vidimo asimetriju podataka i da vizuelno pronadjemo anomalije.
- Scatter plot- pomaze pri detekciji korelacija i da dobijemo generalni osecaj
- Heat map- daje nam vrednost koliko je neka promenjiva povezana sa drugom, bilo bi idealno da nam je neka promenjiva u korelaciji sa dijabetesom, ali to nažalost nije slučaj za naš set podataka. U slučaju da smo imali jaku korelaciju dve promenjive, jednu bi mogli izbaciti.

2.4 Čišćenje podataka

Metode koje sam koristio su:

- Brisanje duplikata, budući da nam je velik skup 3.5% podataka koje izbacujemo nisu bitni.
- Brisanje smeće vrednosti, jedina vrednost je pol koji nije ni muški ni ženski, i budući da tih ima smo 14 vrednosti /100.000, naš model može pogrešno da zaključi.
- Balansiranje podataka, dijagnoza sa dijabetesom ima 10% celog skupa, budući da je to dosta nebalansirano a imamo oko 100.000 podataka dajemo sebi na slobodi

da balanistamo skup metodom “Undersapeling” većinskog skupa kako bi ga izjednačili sa manjinom, izabrran je undersampeling budući da imamo dosta podataka .

- Normalizacija, kako bi svi podaci imali podjedanku važnost pri treniranju, sve brojevne podatke normalizujemo u opsegu [0,1]
- Brisanje anomalija, anomalije koje su preko “brkova” od box plot-a vraćamo na najbližu graničnu vrednost od “brkova”.

2.5 Biblioteke koje su korišćenje

- Sklearn - za modele mašinskog učenja i analizu modela
- Pandas – za čitanje i pisanje CSV-a
- Matplotlib.pyplot - za iscrtavanje rezulatata
- Seaborn - za iscrtavanje rezultata
- Numpy – za rad sa nizovima

3. Modeli mašinskog učenja

1. Modele koje sam koristio

1. Stacking

   - iskoristili smo Random Forest, knn i linearan SVC kao modele izmedju i logističku regresiju koja spaja izlaze iz ta 3 modela i daje jedan izlaz.
   - Ideja je da u ovom slučaju RF, KNN i SVQ istreniramo, ali različiti modeli daju različit izlaz tako da na kraju stavljamo logističku regresiju koja kombinuje izlaze iz 3 modela i daje jedan izlaz (da li pacijent ima ili nema dijabetes).

2. Bagging

   - Uzima nasumično uzorke od trening skupa, deli ih i trenira vise modela paralelno, njihove izlaze spaja kroz metodu većinskog glasnja.
   - Iskoristio sam Decision Tree Classifier.

3. Boosting

   - Radi tako što koristi više modela koji slabo uče u redu i svaki sledeći model služi kako bi ispravio greške prethodnog modela.

4. KNN

- Model koji smo koristili na vežbama, ima lošiji rezultat od ensemble learninga (za 2%).
- KNN je po mojim eksperimentima najmanje otporan na šumove, budući da ga je jedna kolona koja je sadržala vrednosti indexa dovela na preciznost od 52%.
- Takodje sam koristio metodu lakta i zaključio da nema velike razlike greške izmedju 10 i 14 komšija pa sam izabrao 10 radi lakšeg računanja.

5. Stablo odluke
    - Model koji smo radili na vežbama nakon podešavanja hiperparametara imamo 87% uspešnosti.
6. Logisticka regresija
    - Model koji smo takodje radili na vežbama, ima manju uspešnost od kompleksnijih metoda ali ima smisla budući da logističku regresiju koristimo u njima a kod kompleksnijih modela imamo i druge modele koji dopunjuju nedostatke logističke regresije.

2. Izbacivanje parametara

1. Metoda koju sam koristio je bila metoda permutacija, a ona znači da svaki podatak permutujemo i gledamo koliko se menja preciznost modela, ako se predviđanje modela ne promeni, taj parametar nije bitan.
2. Metoda koju sam koristio bila je preko Random forest classifier-a, ona ima sposobnost da nam vrati najbitnije karakteristike, i onda sam zadržao prve četiri.
3. Iako se preciznost modela smanjila za vrlo malu vrednost (manje od 1%), značajno smo smanjili dimenzionalnost problema što znači da nam mreža ima bolje performanse.

4. Analiza performansi modela

4.1 cross validation

- Proces kros-validacije (cross-validation) se koristi kako bi se procenila sposobnost modela da generalizuje na nezavisne podatke. U ovom slučaju, koristi se metoda sa pet preklapanja (cv=5), što znači da se podaci dele na pet podskupova. U svakoj iteraciji, jedan podskup se koristi za testiranje, dok se preostalih četiri koriste za treniranje modela. Ovaj postupak se ponavlja pet puta, svaki put sa drugim podskupom kao testnim skupom.
- Koristili smo ceo skup X,y budući da koristimo kros validaciju kao meru ispravnosti naših modela i ne koristimo pri dodatno obučavanju modela.

4.2 F1 score

- Kombinuje preciznost i odziv modela i izražava koliko su u balansu.

4.3 Odziv

- Odziv (recall) u mašinskom učenju meri sposobnost modela da ispravno identifikuje sve pozitivne instance. Izračunava se kao odnos broja tačno predviđenih pozitivnih instanci (TP) prema zbiru tačno predviđenih pozitivnih i pogrešno predviđenih negativnih instanci (FN).

4.4 Tačnost

- Tačnost (accuracy) u mašinskom učenju meri ukupnu preciznost modela, odnosno odnos ispravno klasifikovanih instanci prema ukupnom broju instanci. Izračunava se kao: [zbir tačnih pogadjanja]/[ukupna pogadjanja].

4.5 Log loss

- Log Loss (logaritamski gubitak), takođe poznat kao logistic loss ili cross-entropy loss, je metrička vrednost koja se koristi za evaluaciju performansi klasifikacionih modela, posebno u binarnoj klasifikaciji. Mera koliko su predviđene verovatnoće udaljene od

stvarnih binarnih oznaka (0 ili 1). Niža vrednost Log Loss-a ukazuje na bolju preciznost modela.
- Log Loss kažnjava velike greške više nego male greške, zbog čega model treba da bude tačan u predviđanju verovatnoća.

4.6 Jaccardov index

- Jaccardov indeks, takođe poznat kao Jaccardova sličnost ili Jaccardov koeficijent, koristi se za merenje sličnosti i raznolikosti skupova. U kontekstu mašinskog učenja, posebno u binarnoj klasifikaciji i segmentaciji slika, meri sličnost između predviđenih i stvarnih binarnih oznaka.
- Jaccardov indeks varira između 0 i 1, gde 1 označava savršeno preklapanje predviđenih i stvarnih instanci, dok 0 označava potpuno različite skupove. Viši Jaccardov indeks označava veću sličnost između skupova.

4.7 Matrica konfuzije

- Matrica konfuzije je alat koji se koristi za evaluaciju performansi klasifikacionih modela, posebno kada se radi o binarnoj ili višeklasnoj klasifikaciji. Ona daje detaljan uvid u to kako model klasifikuje primere u svaku kategoriju, omogućavajući identifikaciju grešaka koje model pravi.
- Za binarnu klasifikaciju, matrica konfuzije je 2x2 i u mom kodu je ispisana za svaki model.
- Kad bi birali matricu kofuzije, za dijabetes bi bilo najbolje da imamo više false negative nego false positive.

4.8 ROC kriva i AUC

- ROC kriva je alat koji ilustruje performanse klasifikacionog modela u različitim pragovima odlučivanja. Ona pruža grafik koji prikazuje odnos između stope lažno pozitivnih (FPR) i stope istinito pozitivnih (TPR) klasifikacija
- AUC (Area Under the Curve) je mera koja odražava površinu ispod ROC krive. Ona pruža agregiranu ocenu performansi modela na svim mogućim pragovima odlučivanja. Veće vrednosti AUC-a ukazuju na bolje performanse modela, gde vrednost 1 označava savršen model, dok vrednost 0.5 ukazuje na slučajnu klasifikaciju. ROC kriva i AUC su korisni alati za analizu i poređenje performansi klasifikacionih modela, posebno kada su klase neuravnotežene ili kada je važno proceniti performanse modela na različitim pragovima odlučivanja.