Detekcija anomalija za pronalaženje prevarnih/lažnih transakcija kod kartica

Petar Marković SW73-2018

Fakultet tehničkih nauka Univerziteta u Novom Sadu

UVOD

Detekcija anomalija predstavlja identifikovanje retkih predmeta, događaja ili zapažanja koji izazivaju sumnje značajnim razlikovanjem od većine podataka. U današnje vreme, neophodno je da banke imaju neki vid zaštite od prevarnih plaćanja, kako bi se maksimalno zaštitile i banke i klijenti. Detekcija anomalija predstavlja jedno dobro rešenje za potrebe banke, uzimajući u obzir da je lažna transakcija sama po sebi neki vid anomalnog predmeta.

DATASET

Za potrebe ovog projekta korišćen je dataset koji sadrži 284807 transakcija od kojih su 492 prevarne. Karakteristike V1, V2, V3.... V28 su glavne komponente dobijene pomoću PCA. Pored ovih karakteristika, postoji još i Time što predstavlja protekle sekunde od svake transakcije i prve transakcije, Amount što predstavlja količinu transakcije, kao i Class što uzima vrednost 1 u slučaju prevare a u suprotnom uzima vrednost 0.

Mora se uzeti u obzir da je ovaj dataset izuzetno neuravnotežen (svega 0.172% transakcija je prevarno). Zbog toga je urađen undersampling prvobitnog skupa podataka, tako da sad postoji 10492 transakcije od kojih je 492 prevarno.

Sam dataset je podeljen na:

- Trening skup – 80%
- Test skup – 20%

PYOD I ALGORITMI

PyOD je skalabilna Python biblioteka za otkrivanje udaljenih objekata u viševarijantnim podacima. Ono što čini PyOD izuzetno korisnim alatom za detekciju anomalija jeste to što sadrži preko 30 različitih algoritama za detekciju. Za svrhu ovog projekta korišćena su 3 algoritma: K-Nearest Neighbors, AutoEncoder I Isolation Forest.

K-Nearest Neighbors

KNN je supervised ML algoritam koji se često koristi za probleme klasifikacije. To je jedan od najjednostavnijih, ali široko korišćenih algoritama sa dobrim slučajevima korišćenja kao što su sistemi preporuka za izgradnju, aplikacije za otkrivanje lica itd.

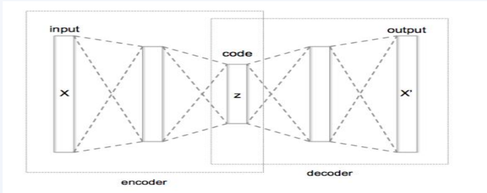
Osnovna pretpostavka je da su slična zapažanja u blizini jedno drugog, a anomalni slučajevi su obično usamljena posmatranja, držeći se dalje od skupa sličnih zapažanja.

Iako je KNN supervised ML algoritam, kada je u pitanju detekcija anomalija, potreban je nenadgledani pristup. To je zato što u proces nije uključeno stvarno „učenje“ i nema unapred određeno označavanje „outlier“ ili „not-outlier“ u skupu podataka, već se u potpunosti zasniva na graničnim vrednostima.

AutoEncoder

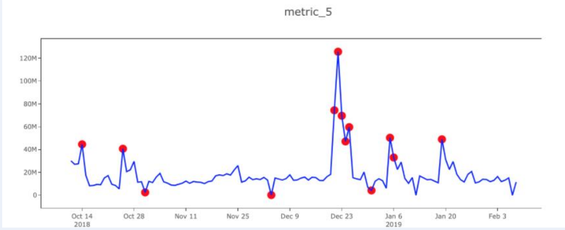
AutoEncoder je unsupervised neuronska mreža koja pokušava da kodira podatke kompresujući ih u niže dimenzije (sloj uskog grla ili kod), a zatim dekodirajući podatke za rekonstrukciju originalnog ulaza. Sloj uskog grla (ili koda) sadrži komprimovani prikaz ulaznih podataka. Broj skrivenih jedinica u kodu naziva se veličina koda.

AutoEncoderi se neretko koriste u otkrivanju anomalija. Greške rekonstrukcije se koriste kao rezultati anomalije.



Isolation Forest

Isolation Forest je prvi algoritam za otkrivanje anomalija koji identifikuje anomalije pomoću izolacije. On uvodi fundamentalno drugačiju metodu koja eksplicitno izoluje anomalije pomoću binarnih stabala, demonstrirajući novu mogućnost brzog detektora anomalija koji direktno cilja anomalije bez procesa intenzivnog profilisanja normalnih instanci.



REZULTATI

Nakon što su svi modeli istrenirani, ovo su rezultati:

Rezultati nad trening skupom

model	ROC	precision @ n	tačno predviđene anomalije	netačno predviđene anomalije
KNN	0.9566	0.5482	372	22
AE	0.9498	0.6731	361	33
IF	0.9563	0.6701	366	28

Rezultati nad test skupom

model	ROC	precision @ n	tačno predviđene anomalije	netačno predviđene anomalije
KNN	0.9393	0.5408	87	11
AE	0.9367	0.5612	86	12
IF	0.9375	0.5714	85	13

Možemo primetiti da su sva tri algoritma izuzetno efikasna, kao i da je K-Nearest Neighbors imao najbolje rezultate i kod trening skupa i kod test skupa.

ZAKLJUČAK

Kao što možemo videti, postignuti rezultati su relativno solidni. Kod trening skupa procenat tačno pogođenih anomalija je ≈ 92%, dok je kod test skupa procenat ≈ 87%.

Iako zadovoljavajući, ovi rezutlati bi se mogli dodatno poboljšati. Najbolji način jeste izmenom parametara naših algoritama. Kod KNN bismo mogli promeniti broj komšija ili metodu za izračunavanje “outlier score”, dok smo kod AE mogli izmeniti sakrivene neurone ili povećati broj epoha.