Primerjava klasifikacijskih modelov na neuravnoteženih podatkih

Ž. Nagelj, L. Lončarič

September 11, 2019

1 Uvod

Cilj naloge je predstaviti in primerjati štiri različnie metode za klasifikacijo na primeru neuravnovešenih podatkov. Gre se za več kot 500 tisoč transakcijskih podatkov različnih tipov na podlagi katerih želimo zaznati prevaro. Za konec bomo v primerjavo dodali še enasamble modelov in sicer v primeru večine glasov ter konsenza.

2 Podatki

Podatke je na spletni strani Kaggle (https://www.kaggle.com/c/ieee-fraud-detection/data) zagotovilo podjetje Vesta. Podatkovni set ima več kot 350 številčnih in kategorični neodvisnih spremenljik. Pomen posameznih spremenljivk ni pojasnjen, so pa definirani naslednji sklopi:

- TransactionDT : timedelta from a given reference datetime
- TransactionAMT : transaction payment amount in USD
- ProductCD : product code, the product for each transaction
- card1 - card6 : payment card information, such as card type, card category, issue bank, country, etc.
- addr : address
- dist : distance
- P and R emaildomain : purchaser and recipient email domain
- C1-C14 : counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked
- D1-D15 : timedelta, such as days between previous transaction, etc.
- M1-M9 : match, such as names on card and address, etc.
- Vxxx: : Vesta engineered rich features, including ranking, counting, and other entity relations

2.1 Nove spremenljivke

Posebno naravo ima spremenljivka *TransactionDT*, ki je periodična. Pomensko nas zanima tedenska perioda, ki jo bomo zajeli tako, da bomo glede na vrednost spremenljivke transakcijo uvrstili v enega izmed sedmih razredov (dni).

2.2 Priprava podatkov

Prvi korak obdelave podatkov je obsegal izbiro kakovostnih spremenljivk glede na delež manjkajočih vrednosti. Odstranili smo tiste spremenljivke, katere delež manjkajočih vrednosti je presegal 20%. Večina odstranjenih spremenljivk je imela oznako V (dodatne spremenljivke, ki jih je merilo podjetje) in D (informacije o časih med transakcijami).

Opazili smo, da pri nekaterih spremenljivkah nastopajo vedno enake vrednosti, zato smo odstranili tudi tiste spremenljivke, katerih varianca je bila praktično nič. Pri tem smo zaradi velikega števila transakcij brez prevar s testom ANOVA preverili, da nizka varianca ni posledica neuravnoteženega podatkovnega seta. Testirali smo torej statistično značilnost razlik med povprečij ob in brez prevare.

Analizo bomo izvajali na popolnih podatkih, torej na tistih brez manjkajočih vrednosti. Po obdelavi nam ostane 346873 transakcij in 96 neodvisnih spremenljivk. Izkaže se, da je količina podatkov prevelika za procesiranje na običajnih osebnih računalnikih, zato bomo s pomočjo stratificiranega vzorčenja vzeli le polovico podatkov, ki jih bomo razdelili na dva dela z namenom nepristranske validacije modela.

2.3 Izbira spremenljivk

Kljub manjšemu številu transakcij imamo še vedno preveliko število neodvisnih spremenljivk. Saj cilj analize ni iskanje čim boljšega modela temveč primerjava različnih metod strojnega učenja. Na podlagi random foresta bomo izbrali 20 spremenljivk, ki pripelje do največje klasifikacijske točnosti. Rezultat v primeru 96 spremenljivk poda točnost modela 0.9644 ± 0.0008 ter v primeru 20 spremenljivk 0.9686 ± 0.0012. Izbrane so bile naslednje spremenljivke:

- TransactionAmt
- card1, card2, card5, card6
- emaildomain
- C1, C2, C6, C9, C11, C13, C14
- V76, V78, V83, V283, V285, V294, V296

2.4 Vizualizacija izbiranih spremenljivk

TransactionAmt

TransactionAmt oz. vrednost prenešenega denarja je edina prava zvezna spremenljivka v naših podatkih. Vidimo, da je spremenljvika približno eksponentno porazdeljena, s barvo, so v vsakem stolpcu označeni deleži transakcij, katere so bile klasificirane kot goljufije. Iz grafa vidimo, da se večina teh zgodi, kadar je vrednost transakcije med 0 in 125.

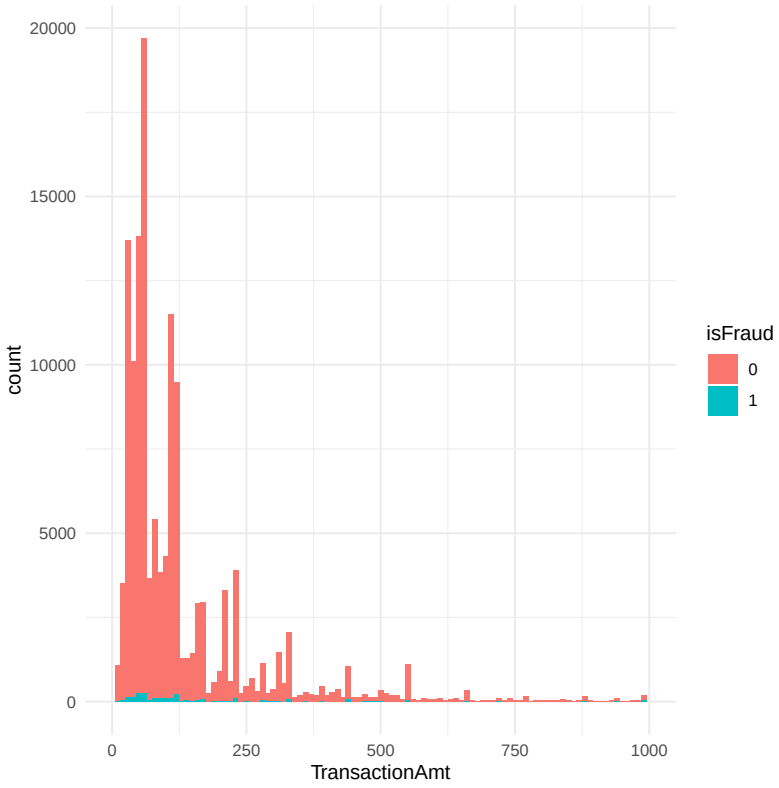


Figure 1: Histogram zneskov transakcij

Card

Spremenljivke tipa *card* skupaj sestavljajo indentifikacijske številke kartic, razen zadnje spremenljivke, ki podaja vrsto kartice. Govoriti o porazdelitvah tukaj nima ravno smisla, saj te vrednosti ne predstavljajo neke zvezne količine. Iz zadnjega grafa vidimo, da imamo v našem vzorcu veliko več transakcij z debitnimi karticami kot pa s kreditnimi karticami.

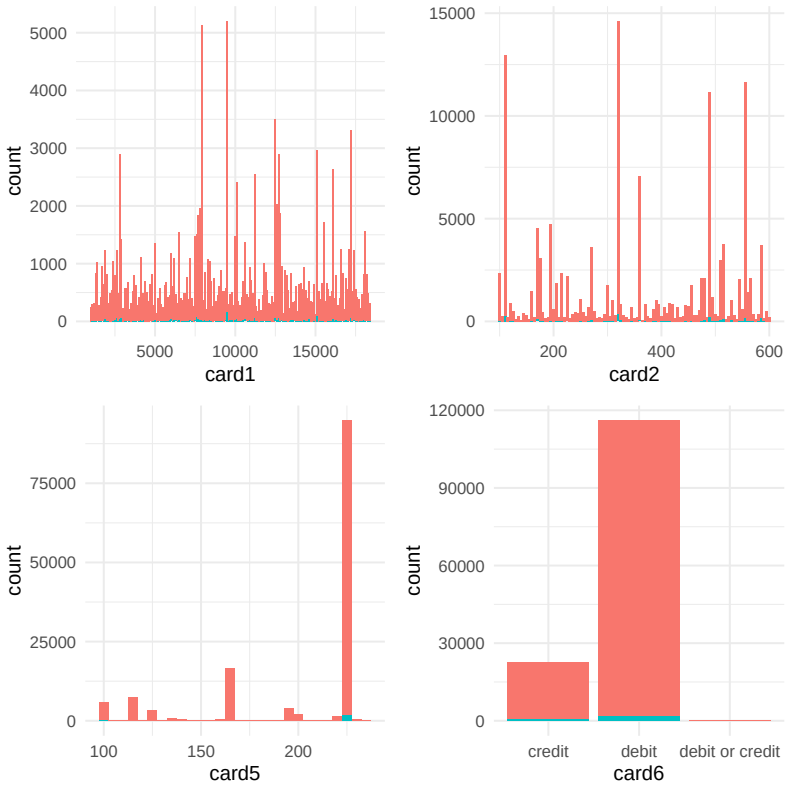


Figure 2: Histogrami spremenljivk kategorije card

Email domain

Email domena je kategorična spremenljivka povezana s transakcijo. Na spodnjem grafu so prikazane le najbolj zastopane skupine. Najbolj zastopana skupina je pričakovano gmail.com, sledi ji pa yahoo.com, ki imata številčno gledano tudi največ goljufivih transakcij.

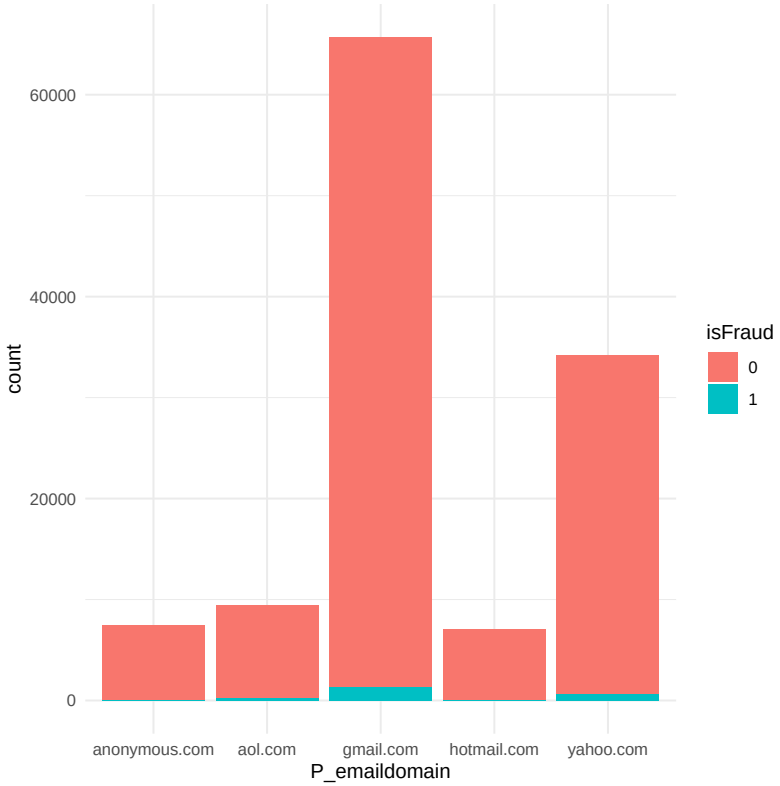


Figure 3: Stolpični diagram email domen plačnika

C

Spremenljivke tipa C so pravtako kodirane spremenljivke, ki naj bi "merile" atribute kot so število asociranih računov s plačilno kartico. Iz normiranih histogramov vidimo, da so vse spremenljivke zelo podobno porazdeljene.

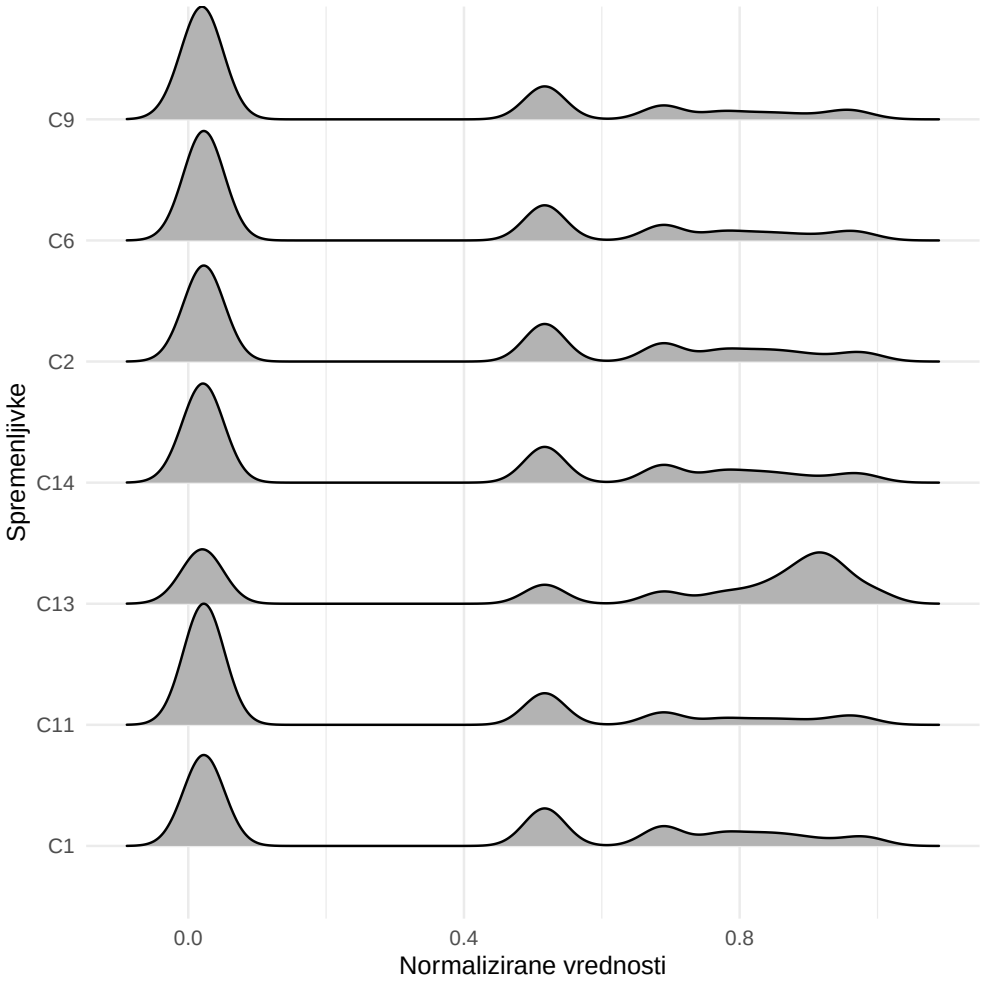


Figure 4: Empiričina porazdelitev spremenljivk kategorije C

V

Spremenljivke tipa V so po meri narejene spremenljivke, ki jih je organizator natečaja ”meril” pri transakcijah. Spremenljivke so pravtako zakodirane, zato ne vemo njihovega pravega pomena. Iz normiranih histogramov vidimo, da so spremenljivke V83, V78, V76 podobno porazdeljene, največ vrednosti se nahaja na sredni. Ostale štiri spremenljivke so tudi medseboj podobno porazdeljene. Iz normiranih histogramov ne moremo razbrati, nobene znane, porazdelitve.

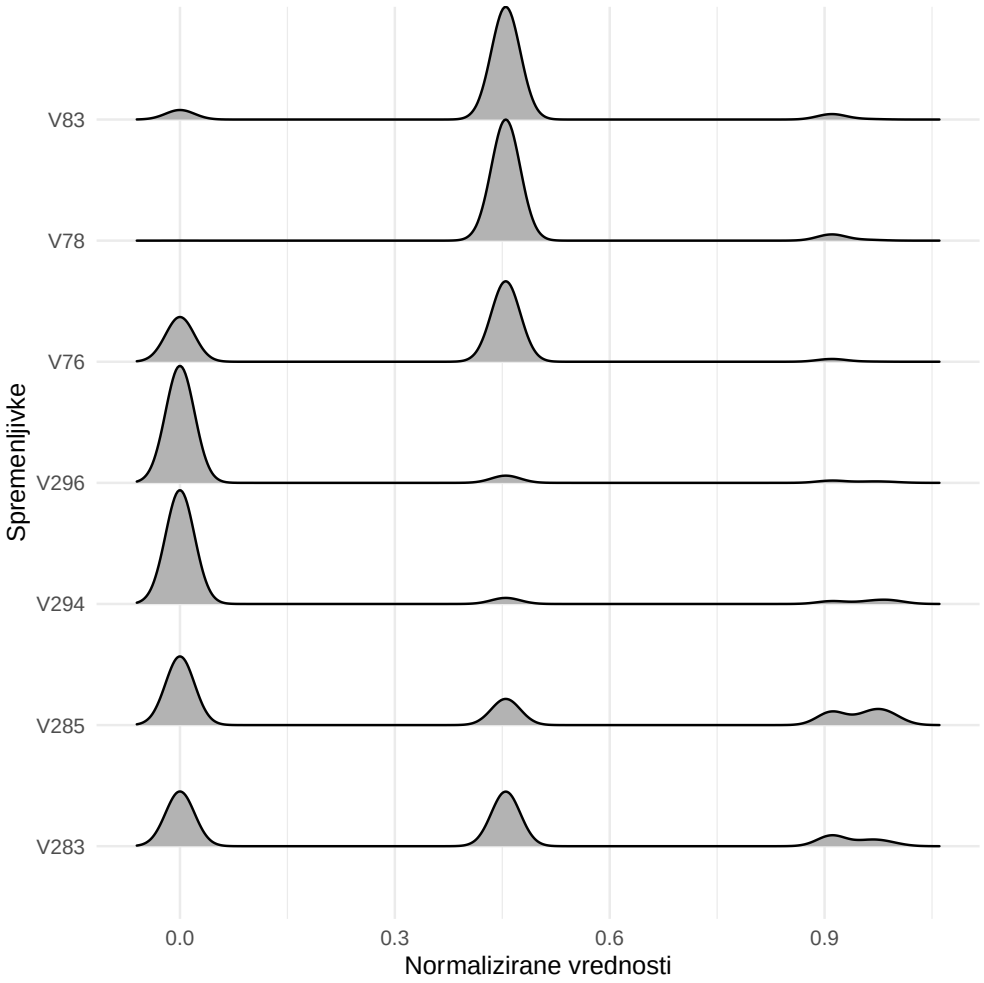


Figure 5: Empiričina porazdelitev spremenljivk kategorije V

Korelacijska matrika

Iz grafa korelacijske matrike vidimo, da so spremenljivke tipa C zelo močno korelirane med seboj, kar lahko nakazuje na probleme s multikolinearnostjo. Spremenljivke tipa V so medseboj šibko korelirane. Ostale korelacije niso značilne.

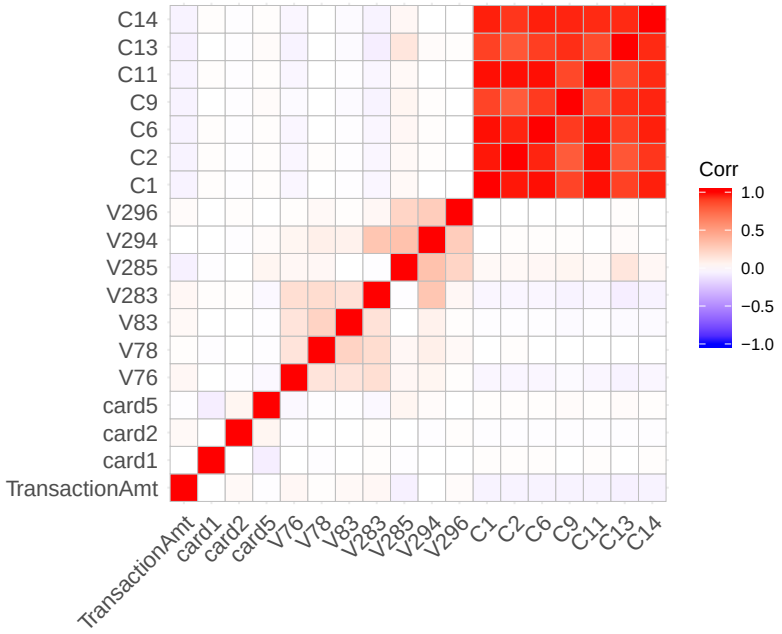


Figure 6: Korelacijska matrika med vsemi izbranimi spremenljivkami

3 Rezultati

Ogledali si bomo rezultate štirih različnih modelov in dveh na podlagi glasovanja. Pri delu z neuravnoteženimi podatki se moramo zavedati zavedljivosti klasifikacijske točnosti. Zato je potrebno, da poznamo dve številki in sicer delež pravilno uvrščenih enot v primeru, da vse enote razvrstimo v skupino 0 (ni prevara) in delež pravilno razvrščenih enot v primeru naključnega razvrščanja ob predpostavki, da so populacijski deleži enaki vzorčnim. Naši referenčni vrednosti sta torej 0.98 in 0.96. V takšnem primeru je pomemben tudi tip napake oziroma ali se gre za False Positive ali False Negative. Zato bomo poleg klasifikacijske točnosti primerjali tudi Precision (manjhna vrednost nakazuje na veliko število False Positive napak) in Recall (manjhna vrednost nakazuje na veliko število False Negative napak). Ogledali si bomo F1 Score, ki združi prej omenjeni metriki. Vsi modeli so bili ocenjeni na standariziranih podatkih, kakovost modela pa je bila vrednotena na ločeni, testni množici podatkov, ki ni bila del eksploratorne analize in ocenjevanja modela.

3.1 Logistična regresija

Logistična regresija je regresijski model, katerega lahko uporabimo za klasifikacijo. Z modelom ocenimo verjenost, da se je nek dogodek zgodil na podlagi danih podatkov. Pri logistični regresiji uporabimo naravni logaritem, da "stisnemo" izhodne vrendosti modela med 0 in 1. Enačbo regresije lahko zapišemo kot:

$$w_0x^0 + w_1x^1 + w_2x^2 + ...w_nx^n = w^Tx = Logit(P(x))$$

Kjer je $Logit(P(x)) = ln(\frac{P(y=1|x)}{1-P(y=1|x)})$ logaritem razmerja verjetnosti, da se je dogodek zgodil deljeno, da se dogodek ni zgodil. Če zgornjo enačbo antilogaritmiramo in izrazimo ven verjetnost, da se je dogodek zgodil, dobimo model logistične regresije oz. logistično krivuljo, ki modelira nelinearno zvezo med napovednimi spremneljivkami in regresorjem:

$$P(y = 1|x) = \frac{1}{1 + e^{-(w^Tx)}}$$

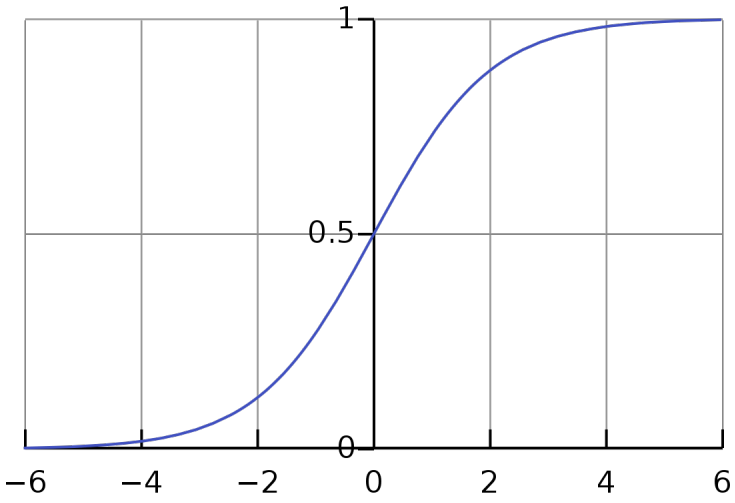


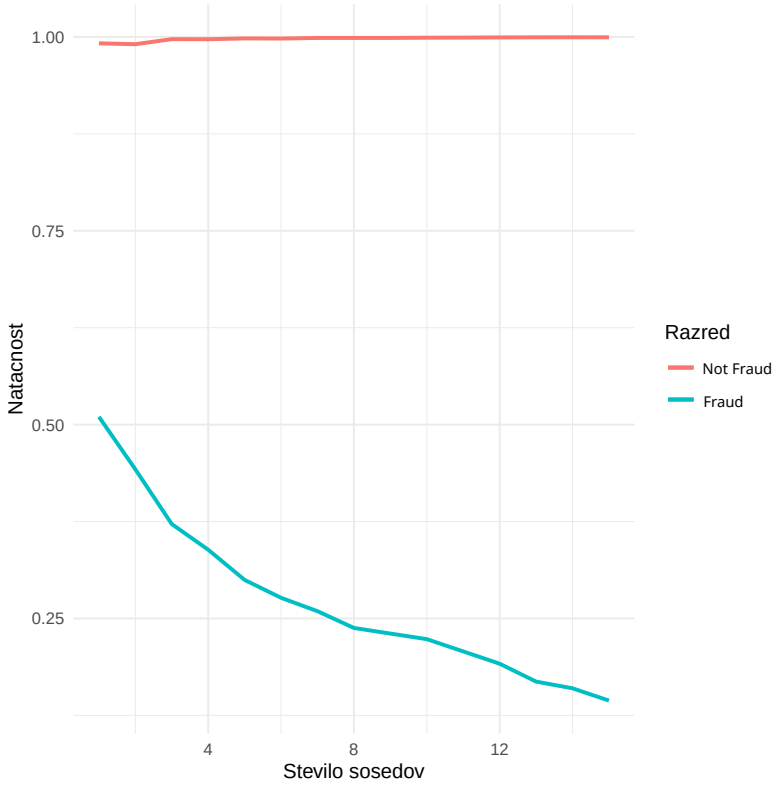
Figure 7: Logistična krivulja

Koeficiente w logistične regresije pa ocenimo tako, da poiščemo take koeficiente, ki maksimizirajo funkcijo logaritem verjetja. To običajno naredimo s numerično optimizacijo kot je gradientni spust. V spodnji tabeli vidimo rezultate logističnega modela, ki je bil natreniran s *glm* ukazom znotraj programa R. Najboljše rezultati so bili doseženi, kadar so bile vse spremenljivke vključene v model. Zaradi multikolienearnosti C spremeljivk smo poskusili vključiti le eno od njih v model, vendar smo v tem primeru dosegeli slabše rezultate. Iz spodnje konfuzijske matrike vidimo, da so rezultati slabi, saj dosegamo slabo natačnost pri klasifikaciji transakcij, ki so goljufive, kar je za nas bolj pomembno kot pravilna klasifikacija transakcij, ki niso goljufive. F1 score je zelo majhen predvsem zaradi majhne vrednosti recalla, torej bo večina napak tipa False Negative.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 33983   679
         1     8    15

               Accuracy : 0.9802
                 95% CI : (0.9787, 0.9816)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : 0.4038

                  Kappa : 0.0406

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.6521739
                 Recall : 0.0216138
                     F1 : 0.0418410
             Prevalence : 0.0200086
         Detection Rate : 0.0004325
   Detection Prevalence : 0.0006631
      Balanced Accuracy : 0.5106892

       'Positive' Class : 1
```

3.2 kNN

K-nearest neighbors oz. K-najbližjih sosedov je ne-parametrična statistična metoda, ki jo lahko uporabimo za regresijo ali klasifikacijo. Pri tej metodi ne ocenjujemo nobenih parametrov, ampak direktno uporabimo podatke za napovedovanje. V našem primeru jo bomo uporabili za klasifikacijo transakcij v goljufive in ne goljufive transakcije. Metoda deluje tako, da transakcijo klasificiramo na podlagi klasifikacij k-tih najbližjih sosedov(transakcij). Najbližje sosede pa določimo, tako da izračunamo razdalje med novim podatkom in vsemi ostalimi podatki po vseh spremenljivkah ali featurjih posebaj, nato pa izberemo k najbližjih sosedov in klasificiramo transakcijo, tako, da ji dodelimo razred, ki je najbolj pogost v množici najbližjih transakcij. Za računanje razdalje običajno uporabimo Evklidovo razdaljo, obstajajo pa tudi druge. Glavna pomankljivost te metode je prav računska zahtevnost, ki raste exponentno z vsakim novim featurjem ali napovedno spremenljivko, kar se je v našem primeru zelo poznalo.
Pri diagnostiki knn modela, si običajno pomagamo s grafom, ki prikazuje natačnost modela v odvisnosti št. sosedov. V našem primeru sem narisal posebaj natačnost za oba razreda, saj je za nas bolj pomebna natančnost pri napovedovanju transakcij, ki so goljufive. Kot vidimo iz grafa, natačnost napovedovanja goljufivih transakcij upada z naraščajočim številom sosedov, zato smo se odličili za $k = 1$.



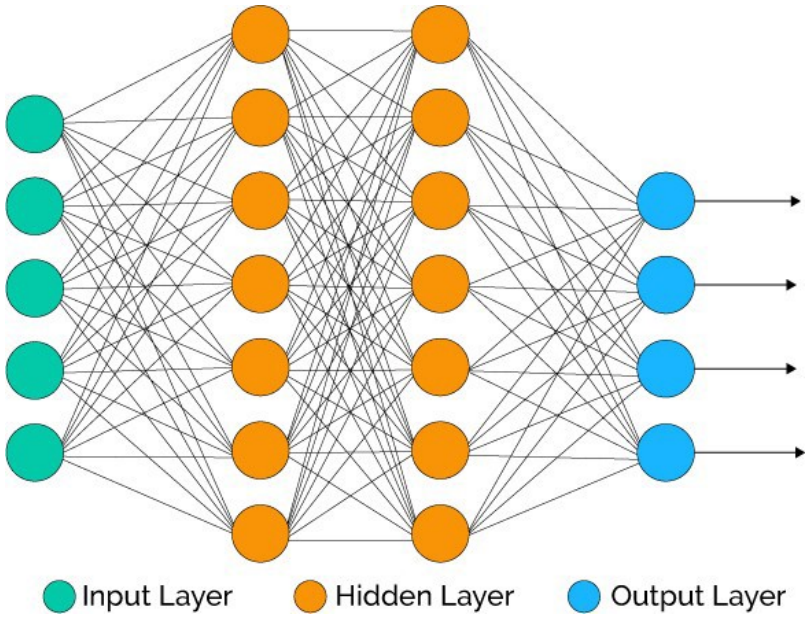
Iz spodnjih rezultatov vidimo, da se je metoda knn odrezala bistveno bolje kot logistična regresija. Odkrili smo za las več kot polovico goljfivih transkacij. Iz tega vidika, je ta model bolšji, kot da bi vzorce razvrščali slučajno. Povprečna natačnost je 0.75, recall in F1 sta tudi visoka kar je dobro. Vendar te metrike nam lahko dajejo lažen občutek o kvaliteti modela, saj je večina napak še vedno False Negative, kar si našem primeru klasifikacije goljufivih modelov ne želimo.

```
Confusion Matrix and Statistics

          Reference
Prediction     1     2
         1 33710   340
         2   283   354

               Accuracy : 0.982
                 95% CI : (0.9806, 0.9834)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : 0.003035

                  Kappa : 0.5228

 Mcnemar's Test P-Value : 0.024859

              Precision : 0.9900
                 Recall : 0.9917
                     F1 : 0.9908
             Prevalence : 0.9800
         Detection Rate : 0.9718
   Detection Prevalence : 0.9816
      Balanced Accuracy : 0.7509

       'Positive' Class : 1
```

3.3 Globoke nevronske mreže

Globoke nevronske mreže so večnivojske mreže z več skritimi nivoji, ki omogočajo večjo kompleksnost modela. Nevronske mreže so priljubljen alogritem klasifikacije, katera ideja izhaja iz delovanja možganov. Vsaka posamezna enota nevronske mreže ima v prvi fazi dodeljene poljubne uteži (bodisi naključne ali s pridobljene s pomočjo specifične inicializacije). Vhodni podatki so nato propagirani skozi posamezne nivoje nevronske mreže kjer se na vsakem nivoju ponovljena operacija matričnega množenja z utežmi trenutnega nivoja ter apliciranje aktivacijske funkcije. Vloga aktivacijske funkcije je, da v model vnaša nelinearno preslikavo prejšnega nivoja in s tem omogoča kompleksnejši model. Hkrati zalogo vrednosti preslika na omejen interval, najpogosteje med 0 in 1. Ko enkrat izračunamo izhod vseh nivojev nevronske mreže sledi proces optimizacije kriterijske funkcije. V tem procesu s pomočjo optimizacijskih metod prilagjamo modelske uteži in s tem minimiziramo napako. Minimizacijo napake izvajamo v obratnem vrstnem redu glede na strukturo modela kot je pretok podatkov, torej iz zadnje (izhodne) plasti se pomikamo proti začetni (vhodni) plasti. Pomemben del učenja modela je tudi proces regularizacije s katerim poskrbimo, da pri modelu ne pride do preprileganja. To naredimo tako, da pri kriterijski funkciji dodamo dodaten regularizacijski člen (L1, L2) ali pa z metodo dropout, kjer v postopku učenja ob vsaki iteraciji uteži naključnih nevronom postavimo na nič.

Naš model je sestavljen iz štirih nivojev, vhodni, izhodni ter dva skrita (15, 10, 5, 1). Pri tem se pri vseh nivojih uporabili aktivacijsko funkcijo hiperbolični tangens razen pri izhodni, kjer smo uporabili sigmoidno aktivacijsko funkcijo. Za optimizacijo smo uporabili algoritem **adam**, ki optimizira binarno prečno entropijo. Za regularizacijo smo uporabili postopek dropout pri prvem in drugem nivoju, kjer ob vsaki iteraciji "ugasne" 20% nevronov. Zaradi velikega števila podatkov smo pri procesu učenja uporabili treniranje na manjših delih podatkov (minibatch) velikost 128 transakcij. Število iteracij učenja (epoch) smo nastavili na 100, a smo omogočili možnost zgodnjega ustavljanja v primeru, če se vrednost kriterijske funkcije ne niža več.



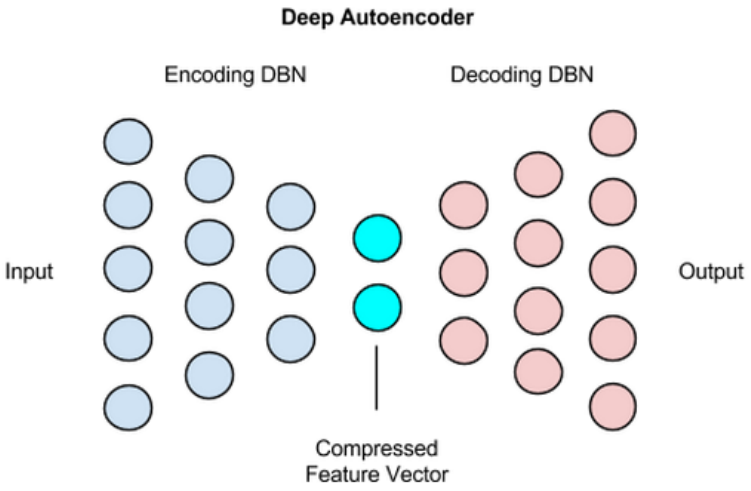
V spodnji tabeli vidimo, da v splošnem klasifikacijska točnost ni boljša kot, če vse enote klasificiramo kot normalne transakcije, je pa boljša v primeru slučajnega razvrščanja. F1 score je zelo majhen predvsem zaradi majhne vrednosti recalla, torej bo večina napak tipa False Negative.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 33988   683
         1     3    11

               Accuracy : 0.9802
                 95% CI : (0.9787, 0.9817)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : 0.389

                  Kappa : 0.0303

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.7857143
                 Recall : 0.0158501
                     F1 : 0.0310734
             Prevalence : 0.0200086
         Detection Rate : 0.0003171
   Detection Prevalence : 0.0004036
      Balanced Accuracy : 0.5078809

       'Positive' Class : 1
```

3.4 Autoencoder

Autoencoderji so nevronske mreže s katerimi se lahko naučimo latentno reprezentacijo (encoding) poljubnega podatkovnega seta. Tradicionalno so bili večinoma uporabljeni z namenom zmanjševanja dimenzij podatkov, trenutno pa so aktulani tudi na področju generativnih modelov. V principu delujejo tako, da skozi plasti nevronske mreže zmanjšamo dimenzijo podatkov (kodirnik), ter nato na podlagi te latentne reprezenzacije vhodne podatke rekonstruiramo s čim manjšo napako (dekodirnik). Da model deluje je potrebna predpostavka, da so porazdelitve spremenljivk transakcij pri katerih je prisotna prevara drugačne od normalnih. Ideja pri uporabi za klasifikacijo pri neuravnoteženih podatkih je naslednja: ker imamo veliko število normalnih transakcij se naučimo latentno reprezentacijo teh. Ko bomo v z modelom napovedovali transakcije, ob upoštevanju predpostavke pričakujemo, da bo napaka pri rekonstrukciji normalnih transakcij manjša kot, ko je prisotna prevara. Določiti moramo še mejno vrednost napake, na podlagi katere bomo klasificirali transakcije. Vrednost napake določimo glede na izbrano metriko, v našem primeru bo to F1.

Naš model je sestavljen iz petih nivojev, vhodni, izhodni ter trije skriti (15, 10, 5, 10, 15). Pri tem se pri vseh nivojih uporabi aktivacijsko funkcijo hiperbolični tangens. Za optimizacijo smo uporabili algoritem **adam**, ki optmizira povprečen kvadrat napake (MSE). Za regularizacijo smo uporabili postopek dropout za nivojema s desetimi nevroni, kjer ob vsaki iteraciji "ugasne" 20% nevronov. Zaradi velikega števila podatkov smo pri procesu učenja uporabili treniranje na manjših delih podatkov (minibatch) velikost 128 transakcij. Število iteracij učenja (epoch) smo nastavili na 100, a smo omogočili možnost zgodnjega ustavljanja v primeru, če se vrednost kriterijske funkcije ne niža več.



V spodnji tabeli vidimo, da je klasifikacijska točnost zelo slaba (0.7049) in ni boljša niti, če vse enote klasificiramo kot normalne transakcije, niti v primeru slučajnega razvrščanja. F1 score je zelo majhen predvsem zaradi majne vrednosti preciznosti, torej bo večina napak tipa False Positive.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 24126   370
         1  9865   324

               Accuracy : 0.7049
                 95% CI : (0.7001, 0.7097)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0229

 Mcnemar's Test P-Value : <2e-16

              Precision : 0.031799
                 Recall : 0.466859
                     F1 : 0.059542
             Prevalence : 0.020009
         Detection Rate : 0.009341
   Detection Prevalence : 0.293758
      Balanced Accuracy : 0.588317

       'Positive' Class : 1
```

3.5 Ensamble

Tak tip klasifikatorja združuje rezultate več različnih modelov. Zdržuili jih bomo na dva načina, gleda na večinski delež glasov (zaradi sodega števila modelov bomo v primeru deleža 0.5 transakcijo klasificirali kot prevaro) ali s konsenzom vseh glasov. V primeru konsenza bomo transakcijo kot prevaro klasificirali le v primeru, če jo za prevaro označijo vsi štirje algoritmi.

Vidimo, da v primeru večinskega glasu je klasifikacijska točnost statistično značilno boljša od referenčne. Večina napak je False Negative. V primeru konsenza klasifikacijska točnost ni statistično značilno različna kot tista, če bi vse enote klasificirali kot normalne. Opazimo, da False Positive napaka ni več prisotna vendar pravilno klasificiramo le eno prevaro.

Večinsko glasovanje

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 33990   473
         1     1   221

               Accuracy : 0.9863
                 95% CI : (0.9851, 0.9875)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4775

 Mcnemar's Test P-Value : < 2.2e-16

              Precision : 0.995495
                 Recall : 0.318444
                     F1 : 0.482533
             Prevalence : 0.020009
         Detection Rate : 0.006372
   Detection Prevalence : 0.006400
      Balanced Accuracy : 0.659207

       'Positive' Class : 1
```

Konsenz

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 33991   693
         1     0     1

               Accuracy : 0.98
                 95% CI : (0.9785, 0.9815)
    No Information Rate : 0.98
    P-Value [Acc > NIR] : 0.4948

                  Kappa : 0.0028

 Mcnemar's Test P-Value : <2e-16

              Precision : 1.000e+00
                 Recall : 1.441e-03
                     F1 : 2.878e-03
             Prevalence : 2.001e-02
         Detection Rate : 2.883e-05
   Detection Prevalence : 2.883e-05
      Balanced Accuracy : 5.007e-01

       'Positive' Class : 1
```

4 Zaključek

Glede na klasifikacijsko točnost je najslabši model autoencoder. Najbolj podobna sta si modela logistične regresije in nevronskih mrež, zelo blizu pa jima je tudi model, ko napovedi določamo s konsenzom, le da je njegov recall precej manjši od prej navedenih modelov. KNN ima najboljše razmerje med preciznostjo in recallom in posledično najboljši F1 score. Zelo blizu mu je tudi, model, ko napovedi določamo na podlagi večinskega glasu. Ta model bi označil tudi kot najboljši, saj edini presega referečno klasifikacijsko točnost, ter ima dobro razmerje med preciznostjo in recallom.

	logistic	knn	neuralnetwork	autoencoder	majority	consensus
Accuracy	0.980	0.982	0.980	0.705	0.986	0.980
Precision	0.652	0.555	0.786	0.032	0.995	1.000
Recall	0.022	0.510	0.016	0.467	0.318	0.001
F1	0.042	0.532	0.031	0.060	0.483	0.003

Table 1: Metrike končnih modelov