# Računarstvo usluga i analiza podataka

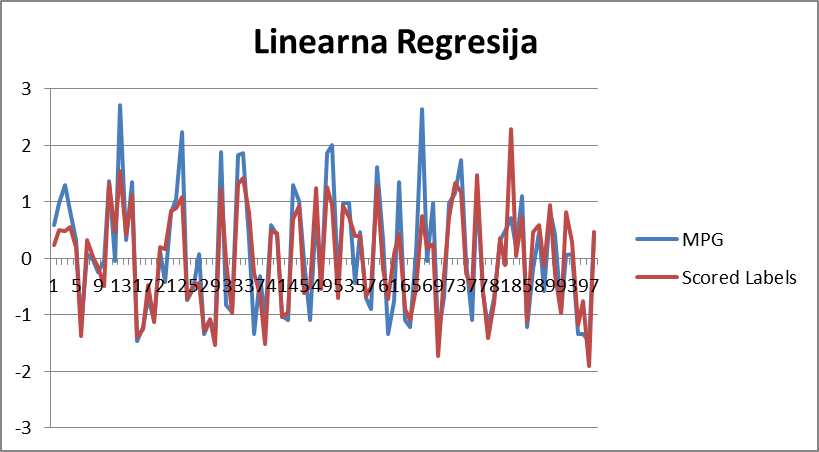
# Laboratorijska vježba 6 :

# Alati i usluge za analizu podataka

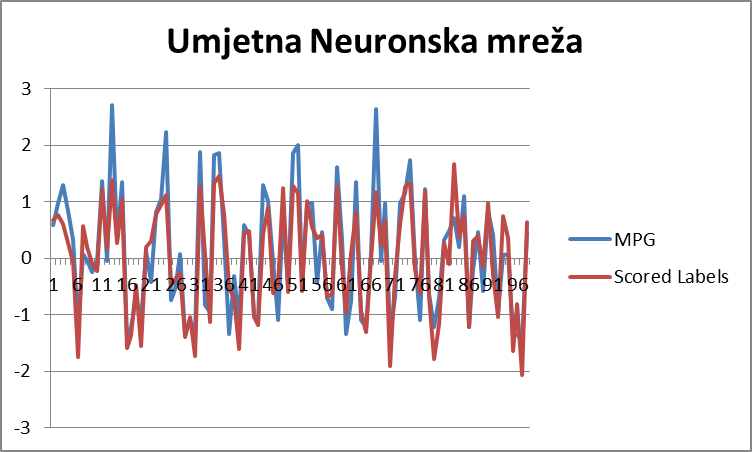
# Izvještaj

## Zadatak 1 :

U prvom se zadatku trebalo složiti dva eksperimenta u Microsoft Azure Machine Leaning Studiu. Navedeno okruženje nam služi za strojno učenje nad ugrađenim podatkovnim skupovima, te naknadno vrednovanje modela. Također omogućuje razvoj izrađenog modela kao servis i korištenje u C# konzolnoj aplikaciji. U prvom zadataku koristio se skup podataka „MPG for various automobiles“. Prvi dio zadatka tražio je od nas da slijedimo predložak i da po njemu napravimo eksperiment. Nad podacima se prvo radila linearna regresija, prije čega je u modelu bilo potrebno očistiti podatke koje nedostaju („Clean missing data“ objekt u Azureu) i normalizirati podatke jer je to pogodnije kad se radi s problemom linearne regresije. Nakon toga smo dijelili podatke u skup za učenje (treniranje) i skup za testiranje u omjeru 75-25% u korist trening skupa. Model se treirao pomoću objekta „Train model“ u Azureu, a u njemu smo odabrali stupac MPG, odnosno odredili smo da su vrijednosti stupca MPG upravo one koje ćemo predviđati u našem modelu. Pokrenuli smo napravljeni eksperiment, nakoon čega smo provjerili rezultate procjene našeg modela u Azure objektu „Score Model“, te smo tablicu usporedbe stvarnih vrijednosti skupa podataka i procjenjenih vrijednosti stupca MPG izvezli kao .csv datoteku u kojoj ćemo grafom prikazati njihove razlike. Također smo u Azure objektu „Evaluate Model“ mogli vidjeti statističke greške između stvarnih i procjenjenih vrijednosti našeg eksperimenta (kao npr. srednja apsolutna pogreška, srednja kvadratna apsolutna pogreška, relativna apsolutna pogreška itd.). U drugom i glavnom dijelu zadatka bilo je potrebno složiti eksperiment gdje smo uspoređivali predikciju linearne regresije i umjetne neuronske mreže (Azure objekt Neural Network Regression). Izvezli smo stvarne i procjenjene vrijednosti u objektu „Score Model“ posebno za rezultate koje smo dobili regresijom neuronske mreže i linearne regresije. U nastavku su prikazani grafovi usporedbe stvarnih i procjenjenih vrijednosti za dva navedena slučaja :



Rezultati dobiveni linearnom regresijom.



Rezultati dobiveni umjetnom neuronskom mrežom.

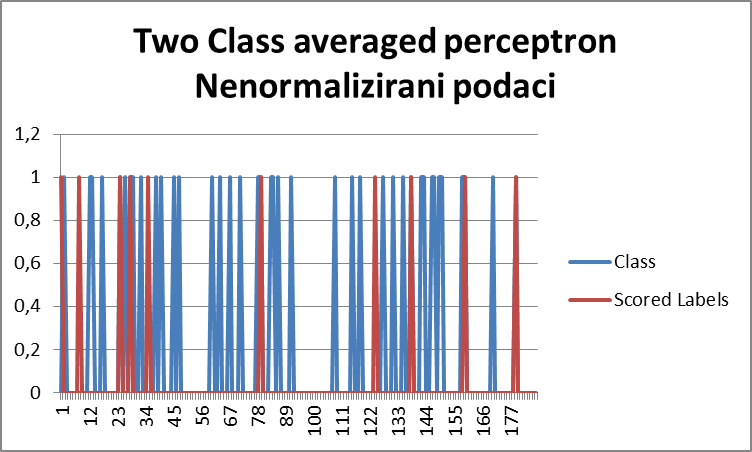
Po prikazanim slikama, vidimo da su linearna regresija i regresija neuronskom mrežom procijelili podatke s približno jednakom točnošću. Ipak, u Azure objektu Evaluate results pokazalo se da linearna regresija ima deset grešaka(41) više od neuronske mreže(31), kao i veće vrijednosti statističkih pogrešaka. Točnije procjenjuje neuronska mreža.

Nakon pregleda točnosti dviju vrsta regresija nužno je razviti (deploy model) istrenirani model. To činimo tako da odaberemo model koji želimo razviti, a mi smo birali onaj koji ima manje grešaka - neuronsku mrežu tako što smo odabrali objekt „Train model“ od neuronske mreže. Pokretanjem novonastalog Predictive Experimenta možemo izvesti eksperiment procjene parametara kao web servis. Taj isti web servis iskoristili smo u C# aplikaciji koju smo kreirali. Web servis nam je dao gotovi kod za slanje zahtjeva na napravljeni web servis. Unošenjem svojih vrijednosti (jedne instance) možemo procjeniti jesu li parametri dobro procjenjeni. Vrijednosti koje smo unosili u C# konzolnu aplikaciju su bile iste vrijednosti pojedinih instanci u korištenom skupu podataka. U kodu je bilo nužno preraditi vrijednost varijable API key u ključ našeg napravljenog API-a koji nam je priložen odmah nakon njegovog kreiranja, kao i dodati funkcije ConfigureAwait() u sve one funkcije koje sadrže await response kako bi mogli vidjeti rezultate nakon pokretanja aplikacije.

## Zadatak 2:

U ovom je zadatku cilj bio usporediti 3 modela klasifikacije podataka : Support Vector Machine binarni klasifikator, Averaged Perceptron binarni klasifikator i Decision Forest binarni klasifikator. Modele smo trenirali i testirali na istom skupu podataka (Blood donation data) s jednakim omjerom trening i test skupa kao u prethodnom zadatku. Eksperiment je trebalo odraditi s nenormaliziranim i normaliziranim podacima skupa podataka. Zadatak svakog klasifikatora je bio razvrstati ljude u dvije skupine : one koji su dali krv i one koji nisu dali krv (u objektu „Train Model“ postavili smo stupac class, vrijednost 1 znači da jesu, 0 da nisu). Support Vector Machine smo koristili s 1 brojem iteracija i lambda vrijednosti 0.001 unutar jednog parametra (class), Averaged Perceptron na learning rate 1 i maksimalni broj iteracija 10, a Decision Forest s 8 stabala odluke. Provjerom u objektima Evaluate Model svakog klasifikatora pokazalo se da su Support Vector Machine i Averaged Perceptron klasificirali podatke s točnošću od 79.7%, dok je Decision Forest bio nešto manje točan – 78.6 % točan. Nakon toga smo razvijali istrenirani model, kad smo razvijali model kao web servis s nenormaliziranim podacima koristili smo model Vector Support Machine binarni klasifikator, a kad smo razvijali model kao web servis s normaliziranim podacima Average Perceptron jer su oba jednako točni (običaj je razviti s najtočnijim klasifikatorom).

Rezultati klasifikacije spremljeni su u .csv datoteke, a sve te tablice su uvežene u jednu xlsx datoteku gdje su se crtali grafovi za svaku izveženu tablicu rezultata. Nacrtani grafovi za normalizirane i nenormalizirane podatke prikazani su u nastavku :



Podaci su jednako točno procjenjeni Averaged Perceptron binarnim klasifikatorom s normalizacijom podataka i bez nje.

Isti slučaj je s Decision Forest klasifikatorom.

Support Vector Machine također pokazuje istu točnost s normalizacijom i bez nje.

Nakon provjere rezultata klasifikacije smo izradili 2 Visual Studio projekta – u svaki ćemo kopirati C# generirani kod napravljenih API-a. Jedan API, odnosno jedan Visual Studio projekt za nenormalizirane podatke razvijen je po Vector Support Machine klasifikatoru, a drugi za normalizirane podatke razvijen je po Average Perceptron klasifikatoru. Pri kreiranju i pokretanju VS projekta (C# aplikacije spojene na API) bilo je potrebno slijediti upute navedene u objašnjenju prvog zadatka. Prilikom pokretanja aplikacije su se također unosile vrijednosti iz početnog skupa podataka (vrijednosti instanci početnog skupa).

Svi Visual Studio projekti (za prvi zadatak i 2 za drugi zadatak) , .csv datoteke s rezultatima predviđanja i klasificiranja dani su u git repozitoriju.