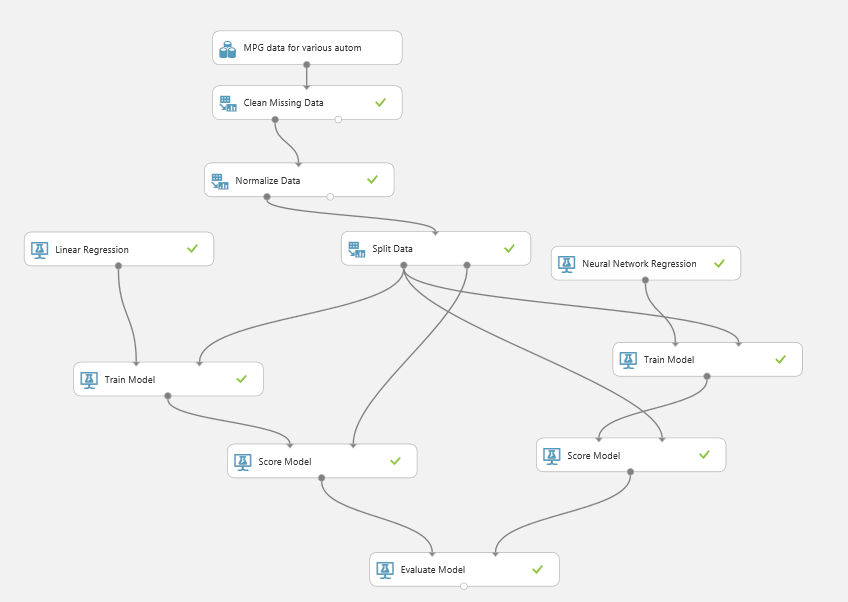
**KOLEGIJ:** Računarstvo usluga i analiza podataka

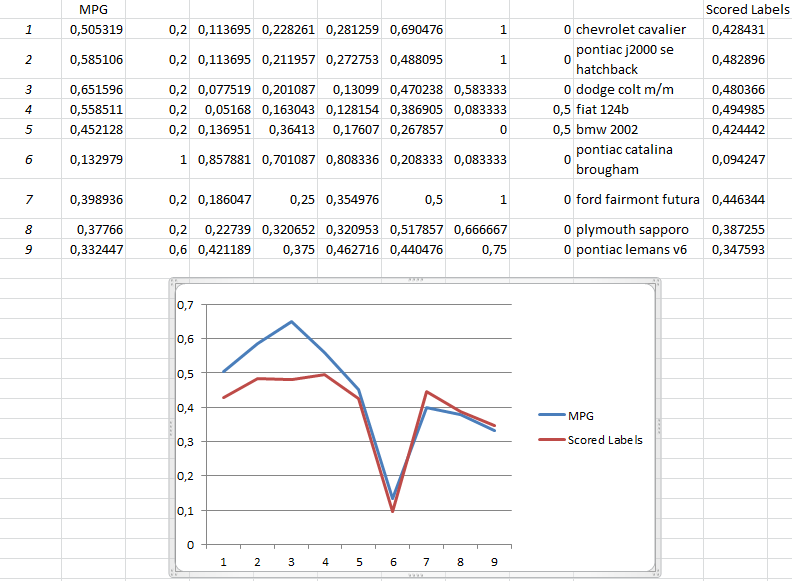
**PREDMET:** Izvještaj sa LV 6

**Zadatak 1.**

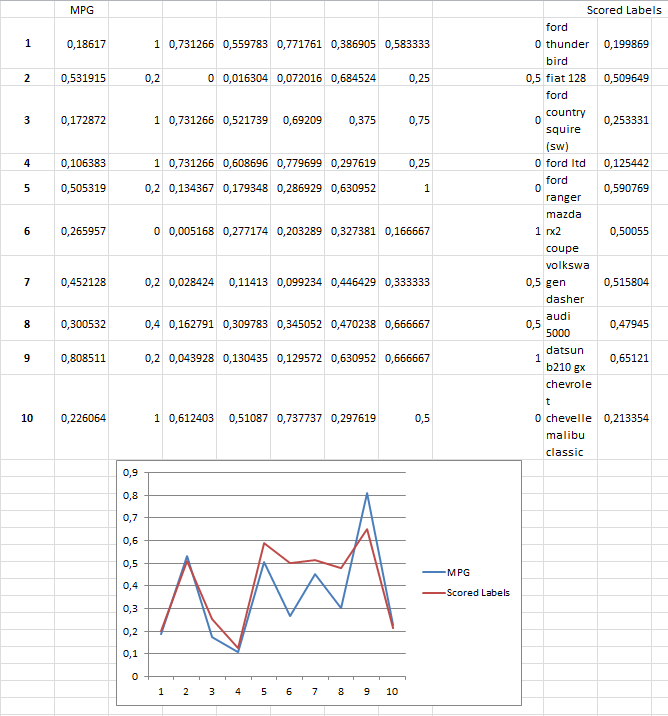
U zadatku1 je potrebno složiti eksperiment zausporedbu postupaka linearne regrecije i regresijske analize korištenjem linearne regresije i umjetne neuronske mreže za regresiju te nakon toga u Excelu generirati linijski graf koji prikazuje vrijednosti MPG, vrijednosti predviđanja linearne regresije te vrijednosti predviđanja umjetne neuronske mreže. Model koji sam koristio za ovaj zadatak prikazan je na sljedećoj slici:



Skup podataka razdjelili smo u podskupove za trenig i testiranje u omjeru 75%:25% a u modelu za testiranje smo odabrali stupac MPG. Zatim smo napravili linijske grafove u Excelu za obje vrste predviđanja a prikazani su na sljedećim slikama:

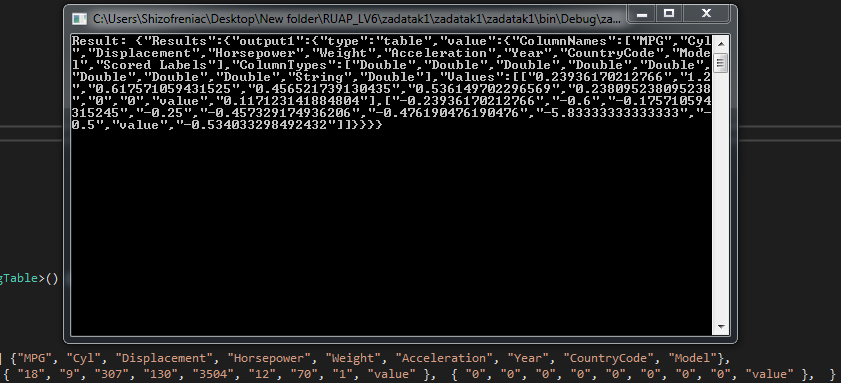


*Vrijednosti linearne regresije za prvih 9 stavaka*



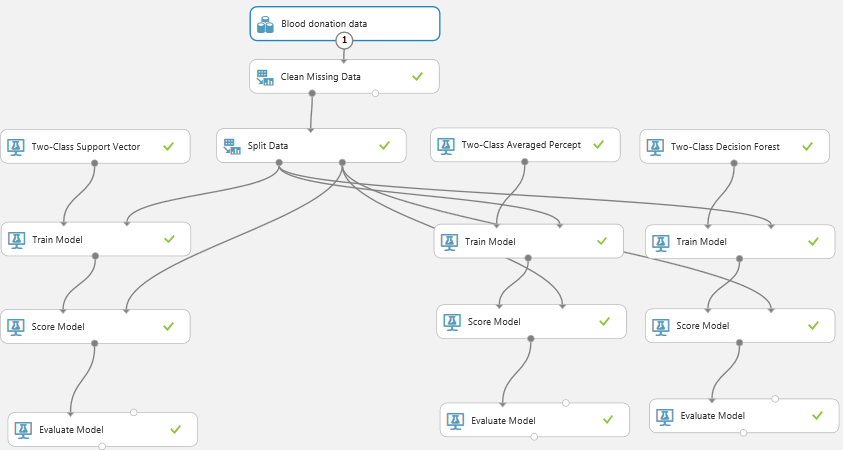
*Vrijednosti umjetne neuronske mreže za prvih 10 stavaka*

Nakon što smo odabrali željeni Train model od istoga je bilo potrebno napraviti Web Service. Nakon što smo kopirali izgenerirani kod te u njega unijeli dodjeljeni API key (OiV6CCOuX27s1A3WdSKXG66rnZsSdA3s5trIAgwmKGyxGJnaig3xHQu5wqYnkG8OeCa19IO2o+L3lh8QLhX1PA==) i instalirali Nuget package Microsoft.AspNet.WebApi.Client aplikacija je bila spremna za upotrebu. Nakon pokretanja aplikacije dobio sam sljedeći rezultat:



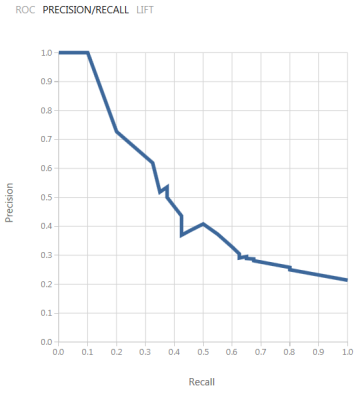
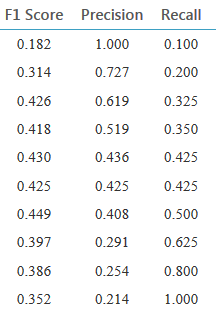
**Zadatak 2.**

U zadatku 2 je potrebno složiti eksperiment za klasifikaciju podataka o darivanju krvi. Za klasifikaciju smo koristili 3 različina modula, Two-Class Support Vector Machine, Two-Class Averaged Perceptron te Two-Class Decision Forest kao što se vidi na sljedećoj slici:

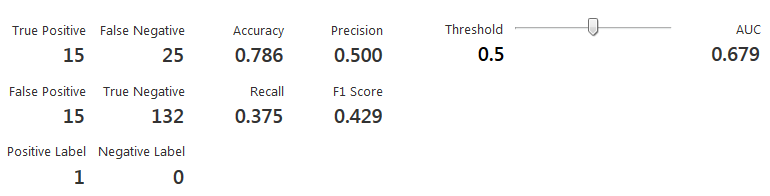


*Model korišten za drugi zadatak*

Nakon što smo složili model prikazan na slici trebali smo odabrati model koji daje najbolje rezultate te na temelju odabranog modela napraviti aplikaciju. Model koji sam odabrao kao najbolje rješenje je Two-Class Decision Forest a odluku sam donio na temelju F1 score rezultata. F1 score računa se formulom F1 score = 2\*(precision \* recall) / (precision + recall), a podaci za navedenu formulu za ovaj primjer prikazani su na sljedećim slikama:



*Precision i Recall vrijednosti za naše vrijednosti*

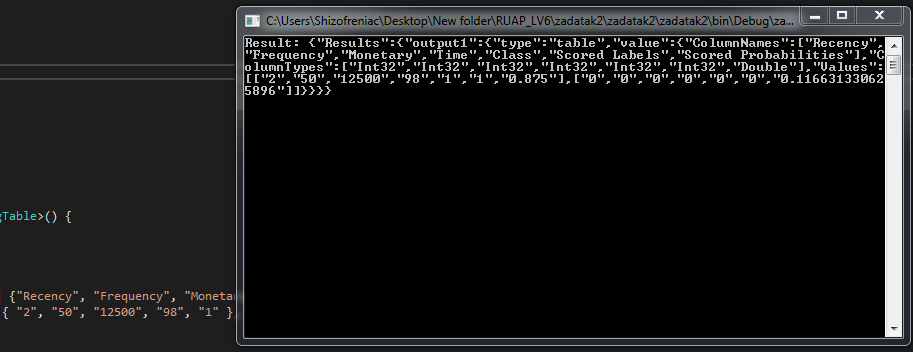


*Konačne vrijednosti*

Kao što je vidljivo na slici F1 score za odabrani evaluacijski model iznosi 0.429 dok ostala dva imaju sljedeće iznose:

* Two-Class Support Vector Machine: F1 Score = 0.240
* Two-Class Averaged Perceptron: F1 Score = 0.240

Nakon što smo odabrali evaluacijski model od istoga je bilo potrebno napraviti Web Service. Nakon što smo kopirali izgenerirani kod te u njega unijeli dodjeljeni API key (J8485fQbWp0yfz1il3P7s0XQq5dSUxS/G6dzmImKdUnsdxDhzg2GgABsafciv4k0CnA8yjiyJ0Mz4eNppWAFSw==) i instalirali Nuget package Microsoft.AspNet.WebApi.Client aplikacija je bila spremna za upotrebu. Nakon pokretanja aplikacije dobio sam sljedeći rezultat:



Svaki korak tijek riješavanja ova 2 zadatka commitan je na githubu kako je i traženo u predlošku ove LV. Na kraju sam merge-ao master i development granu tako da je master grana potpuno ažurirana.

**Vanja Matoković,**

**Osijek, 26. svibnja 2017. godine**