**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

**PREDVIĐANJE USPJEHA UČENIKA**

**Raspoznavanje uzoraka i strojno učenje**

**Ivan Jakab**

**Osijek, 2019. godina**

1. **UVOD**

[[1]](#footnote-1)Cilj seminara je istrenirati različite regresijske modele kako bi predvidjeli uspjeh učenika na kraju godine, na osnovu podataka koje su dali. Modeli će biti istrenirani na skupu podataka koji se nalaze na uci, [Student Performance Data Set](https://archive.ics.uci.edu/ml/datasets/Student+Performance). Podaci su prikupljeni na učenicima u Portugalu te se pratila konačna ocjena učenika na osnovu različitih značajki.

Neke od značajki koje se prate su spol, dob, status roditelja, vrijeme utrošeno na učenje, izlasci, konzumiranje alkohola, izvannastavne aktivnosti itd. Nakon što se model istrenira na tim značajkama, cilj je izraditi jednostavni web sučelje za unos vlastitih podataka, koje će potom dati rezultate koje su predvidjeli različiti modeli.

1. **POSTOJUĆA RJEŠENJA I KORIŠTENE BIBLIOTEKE**

Teško je naći konkretan projekt koji koristi baš ovaj skup podataka, mada vjerojatno postoji. No, postoji jako puno rješenja sa regresijom općenito.

U pythonu postoji puno biblioteka koji rješavaju probleme regresije. Najpoznatija, koja se koristi u ovome radu je scikit learn. Ona daje izrađene modele (ne trenirane), koji većinom dijele isto sučelje prema van – *fit* metoda za učenje, *predict* metoda za dobivanje rezultata. Osim ovih, nudi još puno metoda za predobradu podataka, evaluaciju modela i slično.

Osim scikit learn, koristi se i biblioteka *pandas*, koja omogućava manipuliranje skupom podataka kroz svoj *DataFrame.* Treba i spomenuti biblioteku *numpy*, koju koriste ostale u pozadini, a i u seminaru se koriste radi lakšeg odrađivanja matematičkih operacija.

1. **IZRADA REGRESIJSKOG DIJELA**
   1. **Učitavanje skupa podataka**

Za učitavanje i dohvaćanje podataka, izrađen je poseban modul *dataset.py*. Prilikom inicijalizacije, učitava podatke iz .csv datoteke u pandas DataFrame. Nakon učitavanja, čisti nepotpune podatke (one gdje nisu svi stupci popunjeni). Zatim, odrađuje predobradu koristeći model koji će kasnije biti obrađen. Konačno, razdvaja podatke na one za trening i za testiranje, te osigurava da svi modeli koriste iste podatke.

Prema van, ovaj modul daje metode za dohvaćanje testnog, trening i cijelog skupa podataka. Kod modula je sljedeći:

  
Slika Dataset modul

* 1. **Predobrada podataka**

Nakon učitavanja podataka, slijedi njihova predobrada. Za ovu svrhu je također napravljen poseban modul, *preprocess.py*. Ovo je izdvojeno od dataset modula jer će jednaku predobradu morati proći podaci za učenje i testiranje, kao i podaci pristigli sa web servera.

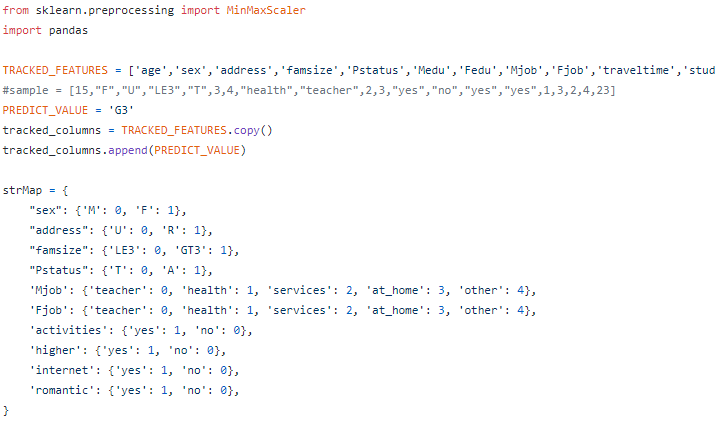
Ovaj modul prema van daje metodu *preprocess*, koji uzima dataframe podaka te ih obrađuje za ulaz u modele.

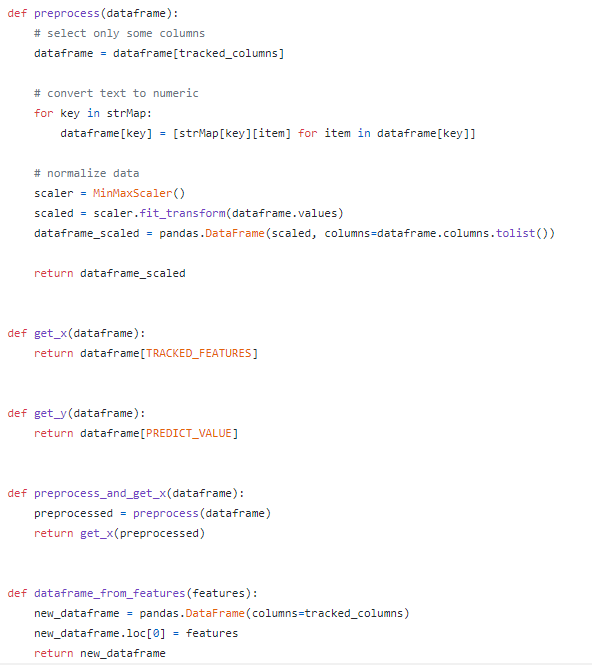
Prvi je korak selekcija značajki koje se prate, jer neke nisu primjenjive u ovome slučaju (npr. škola, koja predstavlja neku školu u Portugalu). Ovo ujedno i osigurava da se ne uzmu svi podaci sa web servera, jer nikada ne možemo znati što će klijent poslati.

Zatim, slijedi pretvaranje riječi u brojeve – skup podataka dolazi sa mnogim riječima koje modeli ne mogu razumjeti (npr. *teacher*), pa ih je potrebno prebaciti u brojeve. Ovo je napravljeno dinamičkim mapiranjem podataka, konkretna implementacija se može vidjeti u kodu.

Na kraju, slijedi korak skaliranja. Za ovo se koristi MinMaxScaler, koji dolazi sa scikit learn bibliotekom. U ovome koraku će se različite vrijednosti numeričkih podataka koji imaju različite raspone (npr. 1-5 konzumacija alkohola, ali 0-93 za izostanke) skalirati na vrijednosti od 0 do 1. Ovim se korakom znatno pojačava preciznost istrenirah modela.

Osim ove metode, modul daje metodu get\_x i get\_y (razdvajanje DataFramea na ulaz i izlaz modela) te neke metode koje će biti korisne kada se uključi web server.

  
Slika Modul za predobradu interno

  
Slika Modul za preobradu sučelje prema van

* 1. **Treniranje i evaluacija modela**

Kao i za ostale stvari, i za ovo je napravljen modul, models.py. Kada se modul inicijalizira, prvo inicijalizira sve modele koji se koriste u seminaru. Njih sprema u mapu, gdje je ključ naziv modela, a vrijednost instanca modela. Linearni modeli koji se koriste jesu Linearna, Bayesova i Lasso regresija. Uz njih koristi se i Random Forest regresija, te na kraju neuronska mreža - Multilayer Perceptron.

Poslije inicijalizacije, dinamički se prolazi kroz sve modele te se na njima poziva *fit* metoda kako bi se model istrenirao. Nakon treninga, modeli daju svoju predikciju za testni skup podataka. Zatim se računa srednja kvadratna pogreška između predviđenih i pravih rezultata testnog skupa.

Rezultati evaluacije su sljedeći

A close up of a logo

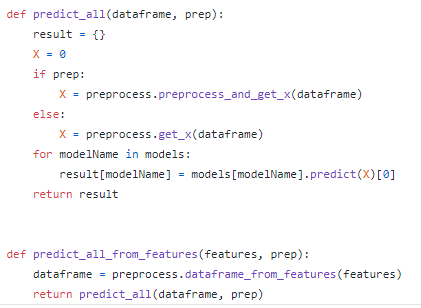
Description automatically generated

Vidi se da najmanju pogrešku ima Lasso regresija, no svi modeli se dobro ponašaju.

  
Slika Modul za modele interno

Prema van, ovaj modul daje metodu *predict\_all*, koja za dani dataframe predviđa rezultat po svim modelima koji su trenirani, te vraća mapu u kojoj je ključ ime modela, a vrijednost predviđeni rezultat. Prije predviđanja će napraviti istu predobradu podataka kao i prilikom učenja modela.

Osim ove, dodana je i metoda *predict\_all\_from\_features*, koja je korisna za web server – uzet će polje značajki, od njega napraviti dataframe i pozvati *predict\_all*.

  
Slika Modul za modele, sučelje prema van

**A map of the computer

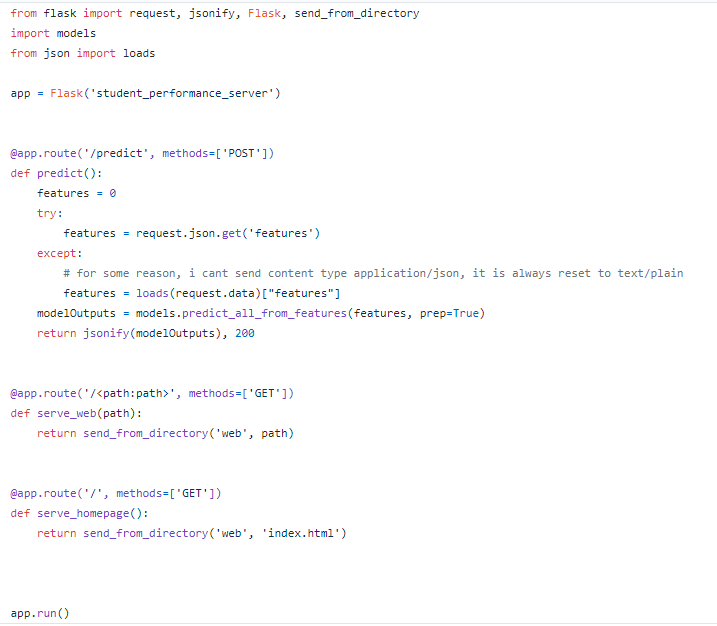
Description automatically generated**Slika Prikaz cijelog procesa, izrađeno u ml studiu

1. **IZRADA WEB APLIKACIJE**
   1. **Poslužiteljska strana**

Kako bi se otvorio jednostavan web server, koristi se biblioteka flusk. On omogućuje jednostavno slušanje na HTTP upite, dobivanje podataka iz upita i slanje odgovora.

Prvo, radi se ruta koja poslužuje statički direktorij. Jednostavno uzme put sa klijenta i posluži datoteku iz direktorija „web“ koja odgovara putu koji je klijent zatražio. Osim ove, postoji i ruta koja poslužuje indeks.html kada nema zatraženog puta.

Posljednja ruta služi za komunikaciju s modelima. Ona je REST tipa, vraća i prima JSON kroz odgovor umjesto neke datoteke. Tipa je post, a očekuje polje značajki. Uzima te značajke poslane sa klijenta te s njima poziva metodu *predict\_all\_from\_features* iz modula za modeleopisanu gore, kako bi dobila rezultat predikcije modela. Taj rezultat vraća kroz odgovor u JSON obliku.

  
Slika Kod poslužitelja

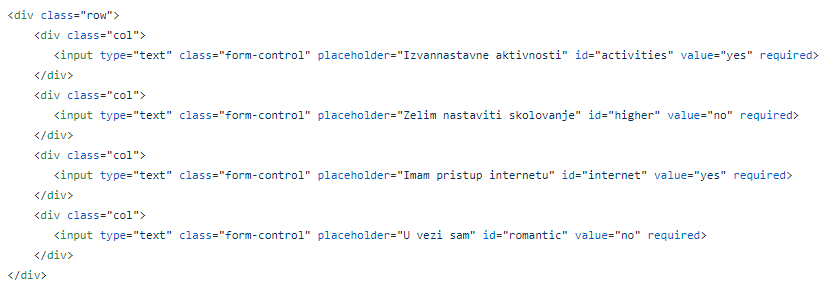
A picture containing screenshot

Description automatically generated  
Slika Prikaz web servisa, izrađeno u ml studiu

* 1. **Klijentska strana**

Na klijentskoj strani, koriste se standardne tehnologije, HTML, CSS i JavaScript. Od biblioteka koristi se bootstrap za brže stiliziranje i chart.js za prikaz grafova. Sav JavaScript je napisan u *script.js* datoteci, a css u *style.css*

U datoteci *index.html* koristi definiran je markup stranice, te su učitane biblioteke, stil i skripte. Većina markupa definira formu s poljima za unos značajki. Značajke imaju jednaka imena kao u dataframeu. Kada se forma podnese, okida se upit na gore opisani poslužitelj, te se pomoću biblioteke iscrtava graf koji prikazuje predikciju različitih modela.

  
Slika Primjer markupa za značajke

JavaScriptom se sluša na predaju forme. Kada se to dogodi, čitaju se unesene vrijednosti značajki. Od njih se izrađuje polje značajki. S tim poljem se poziva poslužiteljeva ruta za komunikaciju s modelima, a poslužitelj vraća u odgovoru vrijednosti koje su pojedini modeli predvidjeli na gore opisan način.

  
Slika Slanje upita na poslužitelj

Kada su predikcije dostupne, pomoću *chart.js* biblioteke se iscrtava graf predviđenih vrijednosti u obliku bar charta.

  
Slika Iscrtavanje grafa

A screenshot of a cell phone

Description automatically generated  
Slika Prikaz web sučelja

1. **ZAKLJUČAK**

U radu je prikazan klasični problem regresijske predikcije. Istrenirano je i evaluirano više različitih modela, te je za svaki prikazan kako se ponaša. Modeli su istrenirani sa relativno malom srednjom kvadratnom pogreškom.

Dodatno, izrađen je web servis kao primjer kako modeli mogu jednostavno komunicirati sa vanjskim svijetom. Također je omogućen grafički prikaz i usporedba različitih modela kako bi se uočile razlike na ovome skupu podataka.

1. P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUture BUsiness TEChnology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7. [↑](#footnote-ref-1)