Analiza projekta iz Strojnog učenja

Mislav Kocijan, Ivan Leverić, Mateo Martinjak

Matematički odsjek, Sveučulište u Zagrebu

26. lipnja 2021.

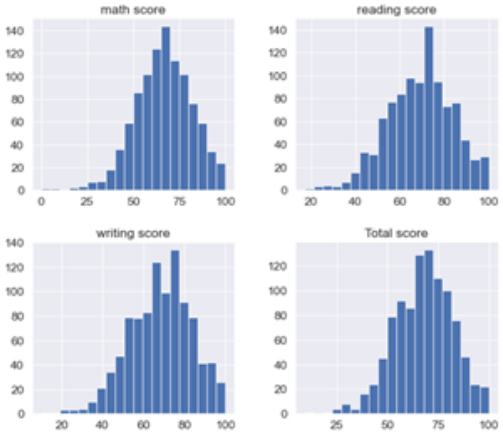
U radu vam predstavljamo rezultate projekta u sklopu kolegija Strojno učenje na Prirodoslovno matematičkom fakultetu u Zagrebu. Obrađena tema se bavi usporedbom modela koji predviđaju uspjeh studenata na ispitima.

Sadržaj

1 Uvod

Promatramo utjecaj odabranih atributa studenata na njihov rezultat na ispitima. Želimo naučiti prognozirati uspjeh studenata na testovima iz pojedinih disciplina, na temelju njihovog spola, rase, dosad stečenog stupnja obrazovanja i slično. Data set koji koristimo je tablica anonimnih studenata u kojem za svakog piše redom: spol, rasa, stupanj obrazovanja, kvaliteta ručka, pripreme prije polaganja, te rezultati iz triju ispita, matematike, čitanja i pisanja,objavljeno na Kaggleu [9] .

Ovdje možemo pisati neki tekst (opis slike) koji će uvijek ići skupa sa slikom.

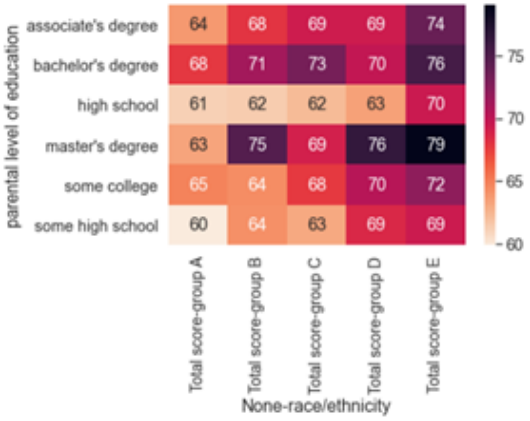


Slika 1: Histogram bodova
Također i ovdje.

2 Opis problema

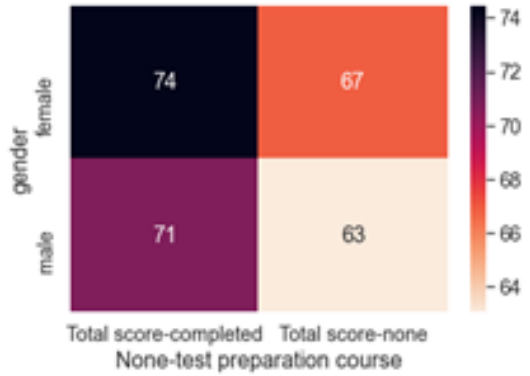
Cilj nam je pronaći (naučiti) funkciju koja za dani niz ulaznih podataka za nekog studenta dati dobru predikciju koliki broj bodova bi taj ili takav student mogao ostvariti na ispitu. Naš skup podataka sadrži informacije o 1000 studenata i studentica. Budući da želimo učiti funkciju koja iz prvih 5 stupaca prognozira preostala tri (ili samo zbroj preostala tri), a ta tri stupca poprimaju samo diskretne vrijednosti, i to prvi 2 (spol), drugi 5 (rasna grupa), treći 5 (stupanj obrazovanja) te četvrti što znači da svih mogućih kombinacija ovih atributa ima konačno mnogo (200) i to manje nego što imamo redaka u tablici (odnosno studenata), znamo da sigurno imamo par studenata (točnije, mora postojati bar jedna petorka studenata) koji imaju sve navedene atribute iste. Međutim, oni vjerojatno neće imati iste bodove na ispitima. Imajući to u vidu, htjeli bismo da nam funkcija osim nekakvog očekivanog broja bodova (vjerojatno prosjeka za neki još veći skup studenata) vraća i neku ocjenu greške, to jest očekivano odstupanje od očekivanog broja bodova (opet, vjerojatno varijanca). Na primjer, za dvije hipotetske grupe studenata koje bi imale sve atribute iste osim pripreme, očekivali bi bolji (prosječni) rezultat grupe koja je imala pripreme, ali i da većina broja bodova bude bliska tom prosjeku, odnosno da svi imaju otprilike podjednaki broj bodova, a za grupu koja nije imala pripreme, osim što bi imala manji prosjek, očekujemo i veća odstupanja od tog prosjeka, da bude studenata s jako dobrim rezultatom i jako lošim.

Ovdje možemo pisati neki tekst (opis slike) koji će uvijek ići skupa sa slikom.



Slika 2: Pivot tablica 1
Također i ovdje.

Ovdje možemo pisati neki tekst (opis slike) koji će uvijek ići skupa sa slikom.



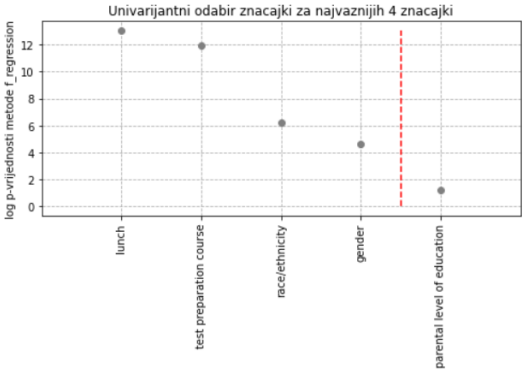
Slika 3: Pivot tablica 2
Također i ovdje.

3 Odabir i encoding značajki

3.1 Encoding značajki

3.2 Odabir značajki

Odabir značajki je biran proces prije učenja jer može značajno ubrzati proces učenja , smanjiti kompleksnost modela (lakše ga je intepretirati) i povećati točnost modela ako se odabere dobar podskup značajki. Mi to ne radimo manualno nego pomoću biblioteke scikit-learn[8]. On sadrži niz metoda za „Feature Selection". Mi koristimo SelectKBest metodu koja će nam vratit k najvažnijih značajki rangirani po bodovima koje određena značajka dobije od algoritma koji proslijedimo kao parametar toj metodi. Mi za regresiju biramo za algoritam f_regression.On je zapravo model za bodovanje koji testira individualni doprinos svake značajke rezultatu.

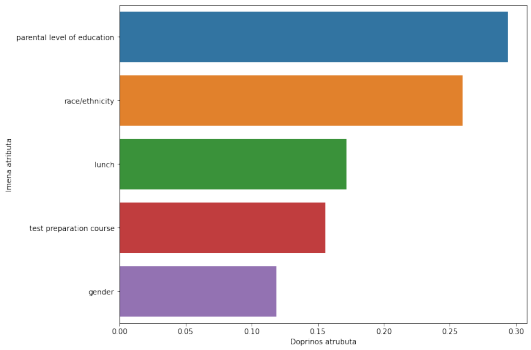


Slika 4: Feature selection using label encoding
Želimo maknuti 20% značajki iz skupa, dakle jednu od pet i testirati oboje.

Algoritam je značajku „parental level of education" stavio na zadnje mjesto pa ćemo testirati kakvu će model imati točnost sa i bez te značajke.

3.3 Embedded feature selection

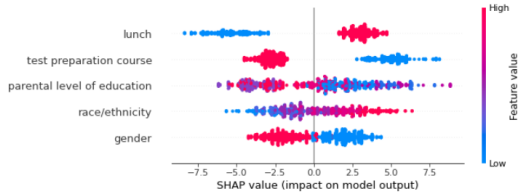
Neki modeli imaju ugrađen „feature importance", tj. mogu nam vratit pojedini doprinos određene značajke u tom modelu.



Slika 5: Doprinos određenih značajki kod random foresta

3.4 SHAP biblioteka

„Predictive models answer the 'how much'. SHAP answers the 'why'." SHAP biblioteka[5] ukratko, objašnjava neki kompleksni model s ciljem da ga lakše intepretiramo. Mi ga koristimo kod LightGBMa.



Slika 6: SHAP- doprinos značajki

4 Opis korištenih modela i mjera

Modeli

- Linearni model
- MultitaskLasso Regresija
- Ridge regresija
- SVR
- kNN
- Slučajne šume

Koristili smo konvencionalne modele poput linearnog i ridge radi potpunosti. MultiLasso regresija[6] je bila zanimljiva u našem slučaju,pošto smo imali 3 izlazne varijable ,te smo ih posebno mogli ubaciti u model. MultiLasso regresija radi upravo to. Kao rezultat dobimo lasso regresiju za svaku izlaznu varijablu posebno.

```
Coefficients:
 [[ 4.14768285  1.77947996 -0.41184818 10.09339002 -4.90915378]
 [-6.06662224  1.08749167 -0.44626441  6.92416825 -6.44344585]
 [-7.68352931  1.39543368 -0.56547683  7.56213732 -8.82239585]]

Intercepts:
 [57.72312274 70.23859011 70.71658567]
```

Slika 7: Rezulati MultiLasso regresije s 3 izlaza

Boosting modeli

Koristili smo dva poznata gradient boosting [1] modela ,prvi je Microsoftov LightGBM,a drugi Yandexov CATBoost.Oba modela obećavaju brže treniranje, bolju točnost , mogućnost paraleliziranja i GPU učenja,kao i učenje nad ogromnim skupom podataka,pa ćemo ih testirati i vidjeti koliko su dobri na našem datasetu.

- LightGBM
- CATBoost

Function	CatBoost	Light GBM
Important parameters which control overfitting	1. **Learning_rate** 2. **Depth** - value can be any integer up to 16. Recommended - [1 to 10] 3. No such feature like min_child_weight 4. **l2-leaf-reg**: L2 regularization coefficient. Used for leaf value calculation (any positive integer allowed)	1. **learning_rate** 2. **max_depth**: default is 20. Important to note that tree still grows leaf-wise. Hence it is important to tune **num_leaves** (number of leaves in a tree) which should be smaller than 2^(max_depth). It is a very important parameter for LGBM 3. **min_data_in_leaf**: default=20, alias= min_data, min_child_samples
Parameters for categorical values	1. **cat_features**: It denotes the index of categorical features 2. **one_hot_max_size**: Use one-hot encoding for all features with number of different values less than or equal to the given parameter value (max – 255)	1. **categorical_feature:** specify the categorical features we want to use for training our model
Parameters for controlling speed	1. **rsm:** Random subspace method. The percentage of features to use at each split selection 2. No such parameter to subset data 3. **iterations**: maximum number of trees that can be built; high value can lead to overfitting	1. **feature_fraction:** fraction of features to be taken for each iteration 2. **bagging_fraction:** data to be used for each iteration and is generally used to speed up the training and avoid overfitting 3. **num_iterations:** number of boosting iterations to be performed; default=100

Slika 8: Važni parametri pojedinog boosting modela

Mjere („metrics") koje smo koristili nad tim modelima su(također u [8] piše detaljnije o njima):

- srednja kvadratna pogreška („MSE") U praksi ovu mjeru koristimo kao mjeru performansi pojedinog modela.Jedna je od načešće korištenih mjera,ali beskorisna ako nam dataset ima puno šuma.Od koristi je kad u datasetu postoji outlieri,ili vrijednosti koje nismo očekivali (da su jako visoke ili jako niske).

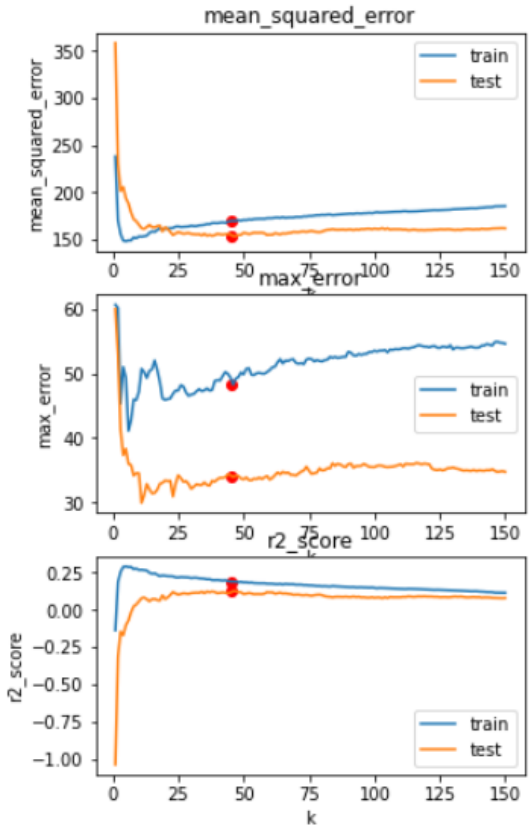
- srednja apsolutna pogreška („MAE") Nije toliko osjetljiv na outliere kao npr. MSE jer ne kažnjava velike greške.Inače se koristi kada se performanse računaju nad numeričkim tipovima podataka,ali smo ju mi koristili radi potpunosti.

- korijen srednje kv. pogreške „RMSE" Pojedinačne pogreške se korjenuju prije nego se sumiraju.To znači da RMSE dodjeljuje veću težinu većim greškama.RMSE je dakle bitan ako očekujemo velike greške u datasetu. Što je RMSE metrika niža to su su performanse modela bolje.

- R^2 ocjena („coefficient of determination") Mjera koja daje ocjenu $\in [0\%, 100\%]$. Ako dodijeli ocjenu 1 to znači da su varijable potpuno korelirane.Niska ocjena znači da postoji relativno niska razina korelacije nad varijablama,znači da je regresijski model slab,ali nemora značiti u svakom slučaju, jer je moguće imati relativno dobar model s lošom R^2 mjerom.

5 Odabir hiperparametara

Hiperparametre biramo ručno, ili nekom formom grid searcha [2], negdje ga implementiramo samo a negdje koristimo gotovu implementaciju. Grid Search radimo na principu da naučimo modele nad određenim skupom kandidata za hiperparametre, i tada gledamo kakva će biti greška nad skupom za testiranje od svakog. Rezultatni model je onaj koji daje najmanju takvu grešku.Pri odabiru biramo za mjeru ili R^2 ili srednju kvadratnu. Može se vidjeti na grafu da nije uvijek da sve 3 mjere daju istog kandidata za model s najmanjom greškom.



Slika 9: Odabir hiperparametra k modela kNN pomoću MSE mjere

5.1 Slučajne šume

6 Kauzalna analiza

6.1 Ispitivanje kauzalnosti pomoću doWhy

Pomoću doWhy library-a možemo ispitati kauzalnost, tj. uzročni utjecaj jedne konkretne varijable na drugu konkretnu varijablu. Općenito, dvije varijable koje su korelirane, mogu biti korelirane jer direktno ovise jedna o drugoj, ili mogu biti korelirani jer postoji neka treća varijabla koja utječe na njih oboje. Library doWhy podrazumijeva da smo za neku varijablu 'uzrok' „posumnjali“ da kauzalno utječe na neku drugu varijablu 'ishod', analizira u kojoj mjeri naš uzrok možda utječe na ishod odnosno, koliko jako su oni povezani, te nalazi varijablu koju bi mogli smatrati zajedničkim uzrokom, ukoliko on postoji, među varijablama koje se navode kao zajednički uzroci ili mogući zajednički uzroci.

7 Rezultati i Zaključak

Pošto točnost naših modela varira oko najviše 50%, tj. model nemože procijeniti bolje od slučajnog odabira, postoji rizik da jednostvano ne postoji veza između naših atributa i target varijable koju smo promatrali.

	MSE	**MAE**	**RMSE**	**R^2**
linearna	141.85	9.46	11.91	0.24
linearna_fs	160.35	9.93	12.66	0.15
Multi-Lasso	162.92	10.42	12.75	0.26
LightGBM	161.46	10.04	12.70	0.14
LightGBM_FS	180.90	10.59	13.45	0.04
kNN	155.01	9.81	12.45	0.18
kNN_FS	172.19	10.43	13.12	0.08
CATBoost	143.15	9.49	11.96	0.24
CATBoost_FS	163.15	10.13	12.77	0.13
RF	177.63	10.56	13.32	0.06
RF_FS	187.10	10.67	13.67	0.01
SVR	142.37	9.47	11.93	0.24
SVR_FS	158.81	9.92	12.60	0.15

[9] [7] [8] [2] [10] [5] [1] [4] [3]

Literatura

[1] *A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning.* URL: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/.

[2] *Bayesian optimization.* URL: https://en.m.wikipedia.org/wiki/Bayesian_optimization.

[3] *CatBoost.* URL: https://catboost.ai/docs/concepts/python-usages-examples.html.

[4] *Causal Inference: Trying to Understand the Question of Why.* URL: https://towardsdatascience.com/implementing-causal-inference-a-key-step-towards-agi-de2cde8ea599?gi=b3bf304d5def.

[5] *Explain Your Model with the SHAP Values.* URL: https://towardsdatascience.com/explain-your-model-with-the-shap-values-bc36aac4de3d?gi=8a24d437e991.

[6] *MultiLasso regression.* URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.MultiTaskLasso.html.

[7] *Other Kaggle projects.* URL: https://www.kaggle.com/spscientist/students-performance-in-exams/tasks.

[8] *Reference guide za cijeli paket scikit-learn.* URL: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics.

[9] *Students Performance in Exams dataset.* URL: https://www.kaggle.com/spscientist/students-performance-in-exams.

[10] *XGBoost, LightGBM or CatBoost — which boosting algorithm should I use?* URL: https://medium.com/riskified-technology/xgboost-lightgbm-or-catboost-which-boosting-algorithm-should-i-use-e7fda7bb36bc.

Popis slika