# UOZP Tekmovanje: Bicikelj

Jan Kuhta[1]

[1]Fakulteta za računalništvo in informatiko, Univerza v Ljubljani

## Analiza in priprava podatkov

V nalogi je bila uporabljena učna množica, ki ima 7739 primerov in 84 atributov, od tega je en atribut časovna značka, ostali pa število koles na posamezni Bicikelj postaji ob tistem času.

Iz atributa 'timestamp' sem zgradil nekaj novih značilk, in sicer 'is_august', ki pove, če je mesec avgust (torej šolske počitnice), 'is_weekend', ki je True kadar je dan v tednu sobota ali nedelja, saj sem predvideval, da so v času šolskih počitnic in ob vikendih toki uporabe koles drugačnih od delovnih dnevov. Iz timstampa sem izluščil tudi zančilko 'hour', ki sem jo one-hot-encodal, torej ustvaril binarni atribut za vsako posamezno uro. Za vsako uro sem ustvaril dve značilki in sicer 'hour_{hour}_week' ter 'hour_{hour}_weekend', saj se uporaba koles precej razlikuje po uri dneva, poleg tega pa se uporaba ob določeni uri razlikuje med delovnimi dnevi in vikendom.

Dodal sem tudi atribut 'is_rainy' na podlagi vremenskih podatkov in ga nastavil na True, če je bila vrednost padavin ('prcp') v tistem dnevu nad 5. Vremenske podatke sem izluščil iz spleta in so shranjeni v datoteki podatki/export.csv. Poizkusil sem tudi z deževnimi podatki po urah ter temperaturnimi podatki, vendar je to model poslabšalo, zato na koncu teh podatkov nisem uporabil.

Med atribute sem dodal tudi podatke o številu koles na vsaki postaji izpred 1h in 2h za modele za napovedovanje za 1h naprej, ter podatke izpred 2h in 3h za napovedovanje 2h naprej. S pomočjo skript closest_1h.py, closest_2h.py in closest_3h.py sem v učni množici poiskal željene podatke za vsako postajo in jih izvozil v csv datoteke, katere sem nato uvozil v glavno skripto in podatke združil z ostalimi atributi glede na njihovo časovno značko. Primere, ki niso imeli podatkov izbred 0-n ur (bodisi začetni podatki, bodisi podatki po testnih luknjah) sem odstranil iz učne množice.

Dodal sem tudi atributa 'is_empty' in 'is_full' ki povesta, če je postaja bila 1 uro prej prazna oz. polna. Podobno sem poizkusil narediti za najbljižjo postajo, vendar to ni izboljšalo modela.

Podatke sem tudi standardiziral in enake koeficiente uporabil tudi pri standardizaciji testnih podatkov.

Za treniranje modela sem uporabil celotno učno množico, na koncu pa naučen model uporabil za predikcijo na testni množici.

## Uporabljen model

Zgradil sem različne regresijske modele in sicer SVM, KNN, nevronsko mrežo, XGBoost, Gradient Boost, Linearno regresijo in Naključni gozd. Vsi modeli so bili zgrajeni na enakih standardiziranih podatkih. Najbolje se je odrezala linearna regresija z regularizacijo (Ridge s stopnjo regularizacije alpha=0.01), ki sem jo uporabil za končno napoved na testnih podatkih.

Gradnjo in treniranje modela sem razledil na dva dela in sicer na napovedovanje za 1 uro naprej in na napovedovanje za 2 uri naprej. Za 1 uro naprej sem poleg osnovnih značilk uporabil podatke o številu koles izpred 1 ure in izpred 2 ur, za 2 uri naprej pa sem uporabil podatke izpred 2 ur in 3 ur.

Gradnjo modelov sem ločil tudi postajah, tako da sem za vsako postajo posebej zgradil ločen model. Odstranil sem vse atribute o številu koles na drugih postajah, tako da so na koncu pstali le osnovni atributi in atributi o številu koles v preteklih urah na postaji, za katero sem gradil model.

Pri napovedovanju na testnih podatkih sem tako ponovno prilagodil učne podatke za vsak model posbej, napovedal število koles za vsako postajo posebej, ter posebej za 1 uro naprej in za 2 uri naprej. Končne podatke sem zaokrožil na cela števila, ter porezal spodnje meje (0) in zgornje meje (glede na maksimalno kapaciteto posameznih postaj).

Na koncu sem podatke združil v končno množico in jo izvozil v končno csv datoteko, ki sem jo nato oddal na spletni strani za oddajo.