Velika domača naloga: koliko bo tam koles?

Gašper Spagnolo[1]

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

Preprocesiranje

Podatki so predprocesirani v dveh korakih. Prvi korak je funkcija *historic data*, drugi korak pa je funkcija *preprocess data*.

V funkciji *historic data*:

- Najprej so uvoženi podatki iz dveh CSV datotek: ena vsebuje zgodovinske podatke o kolesarskih postajah, druga pa razdalje med postajami.
- Nato se vrednosti časovnih znamk v obeh naborih podatkov pretvorijo v format datetime.
- Za vsako postajo se poiščejo tri najbližje postaje.
- Za izbrano postajo se nato v iteraciji zabeležijo zgodovinski podatki (zamik za 30, 60 in 90 minut) o številu koles na postaji in na treh najbližjih postajah. Ti podatki so dodani kot nove značilke v glavni podatkovni nabor.
- Za vsak ˇ casovni zamik se izraˇ cuna skupno število koles za vse postaje in se to vrednost doda kot nova značilka.

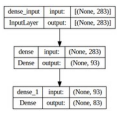
V funkciji *preprocess data*:

- Uvoženi so podatki iz treh CSV datotek: ena za uˇcni nabor, ena za testni nabor in ena za metapodatke.
- Za vsako postajo se izvede naslednja obdelava:
- Iz učnega in testnega nabora se za vsako postajo izločijo podatki o številu koles in ˇcasu.
- Metapodatki se združijo z glavnim naborom podatkov na podlagi časovne znamke.
- Uporabi se funkcija historic data za dodajanje zgodovinskih podatkov.
- Odstrani se časovna znamka iz nabora podatkov.
- Ure, minute in dni v tednu se pretvorijo v kategorične spremenljivke in se uporabi enovitna pretvorba (one-hot encoding).
- Na koncu se vrne seznam postaj, kjer je za vsako postajo shranjena časovna znamka testnega nabora, predprocesirani učni in testni nabori ter ciljne vrednosti za učenje in testiranje.

Treniranje modela(ov)

V okviru projekta sem zaˇcel s pristopom **linearnih regresij** za posamezno postajo. S pomočjo metode najmanjših kvadratov sem prilagodil model za vsako postajo posebej, pri čemer sem se osredotočil na minimizacijo povprečne absolutne napake. V tem procesu sem uspel doseˇci MAE 1,88, kar je bilo obetavno in je postavilo temelje za moje nadaljnje analize.

Nato sem se vprašal, ali bi lahko izkoristil morebitne interakcije med postajami, da bi izboljšal svoj model. Za preuˇcevanje te hipoteze sem se odločil za izgradnjo **nevronske mreže** , saj so te zelo uspešne pri modeliranju kompleksnih interakcij med spremenljivkami. Po skrbnem hipertuningu parametrov sem prišel do zaključcka, da je najbolje uporabiti le en skrit sloj, velikosti 93 nevronov. Topologija mreže je bila torej konfigurirana v obliki 294 -> 93 -> 83, pri čemer sem napovedoval vse postaje hkrati.



Kljub mojemu upanju, da bodo nevronske mreže uspele zajeti morebitne interakcije med postajami, ta pristop ni prinesel želenih rezultatov. MAE, dosežen s to metodo, je bil le 2,4, kar je nekoliko slabše od mojega prvotnega linearnega modela. Nezadovoljen z rezultati moje nevronske mreže, sem se odloˇcil preizkusiti še en algoritem strojnega uˇ cenja: **XGBoost**. To je regresijski model, ki temelji na metodi gradientnega spodbujevanja, znan po svoji učinkovitosti in fleksibilnosti. Na istem naboru podatkov, ki vključuje vse postaje, je XGBoost presegel moj model nevronske mreže in dosegel MAE 2,04. To je še vedno slabše kot moj začetni linearni model, vendar je vsekakor boljše od nevronske mreže.

Skupaj so moje analize pokazale, da preprosta linearna regresija za posamezno postajo v mojem primeru daje najboljše rezultate. Prav tako sem ugotovil, da morebitne interakcije med postajami moji modeli niso uspeli ujeti.

Rezultati

Model	MAE
Linearna regresija	1.88
Nevronska mreza	2.46
XGboost	2.04

Table 1: *Pri linearni regresiji se je uporabilo 83 modelov (toliko kolikor je postaj) pri xgboostu in nevronski mrezi pa je model napovedoval vse postaje.*