Proces

Začel sem z mojim modelom iz prejšnje naloge. Zaradi manjših sprememb v obliki podatkov (ni stolpca »category«, v testnih podatkih ni y, itd.) so bile potrebne minimalne prilagoditve. Ta model je dosegel MAE nekaj več kot 39.
Nadaljeval sem z rekreiranjem baseline modela. Napaka moje verzije je bila malce večja, vendar blizu.
Nato sem preizkušal različne priprave podatkov (različni feature-ji, parametri).
Začel sem s tem, da sem stolpec z avtorji pravilno pretvoril v one-hot obliko. Sprva so bili avtorji zgolj en string, zato sem je bilo potrebno seznam avtorjev posebej kodirati. Na enak način sem potem dodal »gpt_keywords« v one-hot encoding obliki. Z »gpt_keywords« sem opazil slabši rezultat, zato sem to zavrgel in namesto tega uporabil »keywords«, kodirano na enak način. To je izboljšalo rezultat.
Potem sem razširil besedilo. Dodal sem še »lead« polje.
Nato sem besedilo predprocesiral, podobno kot v prejšnji nalogi. Vsa tri polja besedila so najprej konkatenirana, nato pa so pognana čez lematizator, ki podpira slovenski jezik. Potem so besede filtrirane, pri čemer se zavrže stopword-e (slovenske). Vektorizacijo sem pustil enako. To je izboljšalo rezultat.
Poskusil sem one-hot encoder-ju podati parameter »min_frequency«, ki pove, kolikokrat se mora minimalno pojaviti vrednost, da jo encoder uporabi v končni obliki. Že nizke vrednosti drastično zmanjšajo dimenzijo izhodne matrike. TF-IDF vektorizator ima podoben parameter – min_df, ki pove minimalno število dokumentov, v katerih se mora pojaviti beseda. Podano je lahko kot celo število ali delež dokumentov. Tudi tukaj je že nizka vrednost drastično zmanjšala dimenzijo izhodne matrike. Vendar pa sem opazil poslabšanje rezultata v obeh primerih. Ker gre za sparse matrike, ki so zaradi učinkovitosti lahko ogromne, sem se odločil oba parametra opustiti.
Poskusil sem tudi pognati model z drugačnim parametrom alpha – podobnim, kot sem ga uporabil v prejšnji nalogi, vendar je tudi to vodilo do rahlega poslabšanja rezultata, zato sem opustil.
Eksperimentiral sem tudi z zaokroževanjem števila komentarjev na celo število (po potenciranju, ki je potrebno, ker model napoveduje koren št. komentarjev) – št. komentarjev je lahko namreč zgolj celo število. Nisem opazil velike spremembe v napaki.

Priprava podatkov

Iz izvornih podatkov so izluščeni/narejeni naslednji feature-ji:

- **Dan v tednu**, ko je bil članek objavljen (iz datuma)
- **Ura** objave članka (iz datuma, samo ura – brez minut)
- **Minuta** objave članka, zaokrožena na najbližji 15-minutni interval (iz datuma, samo minuta)
- »**topic**« članka (iz »topics«)
- »**subtopic**« članka (iz URL-ja)

Ti feature-i so nato obdelani z one-hot encoder-jem. Rezultat je sparse matrika, kjer so ti feature-ji one-hot encode-ani.
Potem so še **avtorji člankov** posebej one-hot encode-ani. To je narejeno ločeno, ker ima vsak članek seznam avtorjev. Rezultat je sparse matrika.
Enako je narejeno za **ključne besede (»keywords«).**
Naslov (»title«), **uvod/podnaslov (»lead«)** in **besedilo (»paragraphs«)** so za vsak članek konkatenirani. Potem je to besedilo pognano čez lematizator in čez filtriranje stopword-ov. Na koncu pa je vektorizirano z TF-IDF vektorizatorjem. Rezultat je sparse matrika.
Vse 4 sparse matrike so horizontalno zlepljene skupaj. To je X parameter modela.
Parameter y je seznam števil komentarjev, kjer vsako število pripada enemu članku.

Model

Model je linearna regresija z L2 regularizacijo – Ridge. Ta model sem uporabil, ker je bil že pri prejšnji nalogi najbolj obetaven.
Model napoveduje koren števila komentarjev.