Mašinsko učenje – domaći 1

Na početku rada algoritma, potrebno je učitati podatke, skalirati ih i podeliti na trening i test skup. Napisana je funkcija koja nasumično bira odbirke koji će pripasti treningu i testu.

Linearna regresija

Prvi algoritam koji je razmatran je linearna regresija. Moja linearna regresija ima sličnu strukturu kao linearna regresija iz sklearn-a. Ima fit metodu kojoj se prosleđuju vektor prediktora i vektor predikcija. Ovoj metodi se takođe prosleđuju dodatni parametri kao sto su *method* koji određuje na koji način će se proračunavati parametri θ i ima moguće vrednosti *stohastic* i *batch*, na osnovu kojih se primenjuju **stochastic gradient descent** ili **batch gradient descent**.

Ova klasa takodje sadrži metodu **predict** koja ima zadatak da na osnovu ulaznog testnog vektora prediktora proračuna predikciju y.

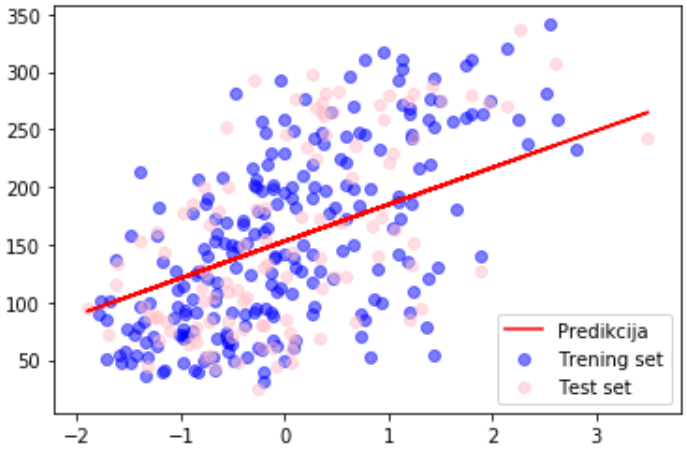
Za poređenje sa ugrađenim funkcijama implementirana je funkcija *my_mse* koja racuna srednju kvadratnu grešku predikcije.

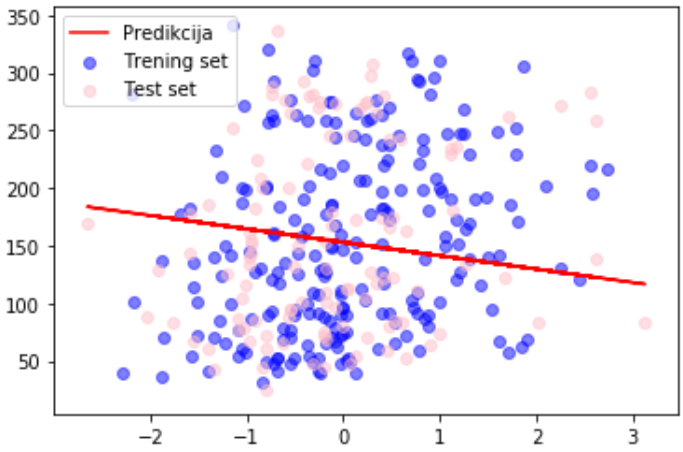
Linearna regresija iz sklearn-a ima mse: 3738.5318263363815

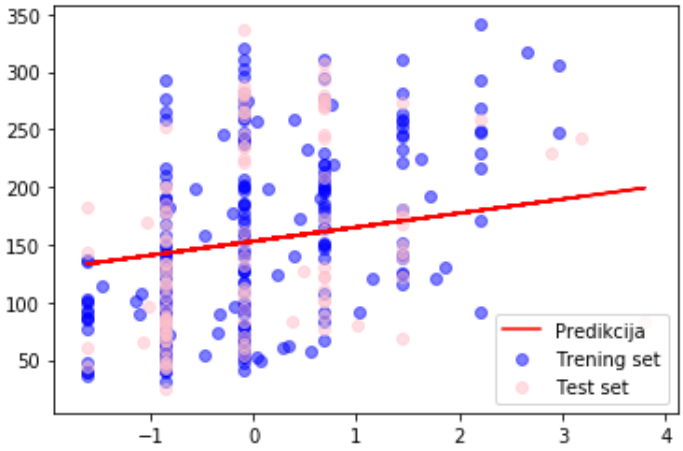
Moja linearna regresija sa batch gradient descent-om: 3738.5318281260443

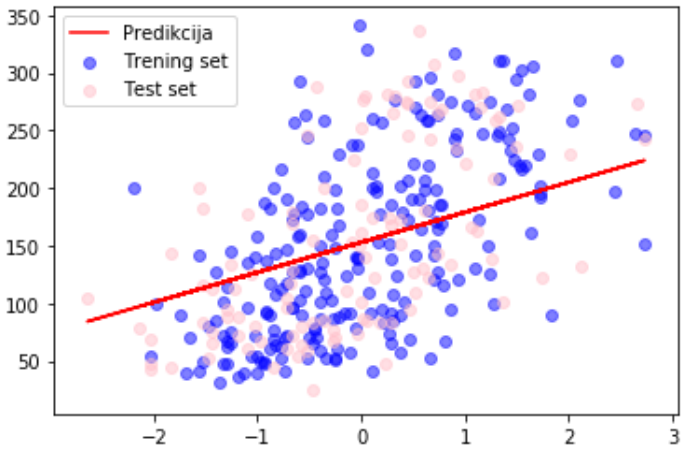
Moja linearna regresija sa stochastic gradient descent-om: 3726.0359468764186

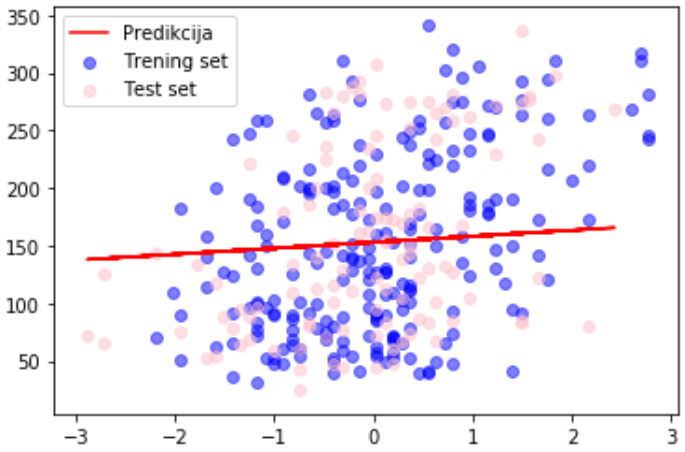
Predikcija za stochastic gradient descent i svaki feature posebno prikazani su na sledećim slikama:




Predikcija
Trening set
Test set
350
300
250
200
150
100
50
−2
−1
0
1
2
3


Predikcija
Trening set
Test set
350
300
250
200
150
100
50
−1
0
1
2
3
4


Predikcija
Trening set
Test set
350
300
250
200
150
100
50
−2
−1
0
1
2
3



S obzirom da svi plot-ovi izgledaju slično za različite algoritme, u ovom izveštaju biće prikazani samo plot-ovi za Linearnu regresiju, dok se ostatak može videte u jupyter notebook-u.

Ridge regresija

Nakon obične linearne regresije implementirana je jedna od njenih regularizovanih varijanti. Parametri θ koji služe za računanje predikcije u ovom slučaju mogu se izračunati u zatvorenoj formi na sledeći način:

$$\theta = (X^T X + \alpha\Lambda)^{-1} X^T y$$

Ugrađena ridge regresija: 3737.972731398663

Moja ridge regresija: 3738.572981342715

Lasso regresija

Lasso regresija se ne razlikuje mnogo od linearne, u implementaciji je u formuli za batch gradient descent dodat član koji se odnosi na znak prethodnog *theta*.

Ugrađena lasso regresija: 3738.4403619354352

Moja lasso regresija: 3737.1494834846544

Lokalno ponderisana linearna regresija

S obzirom da je ovo neparametarski model, njegova implementacija se malo razlikovala od ostalih modela. U ovom slučaju nije bilo podele na trening i test skupove, već se pri ''treniranju'' koriste svi odbirci X osim i-tog za koji želimo da izračunamo predikciju. Pored parametra *theta* sada se računa i novi parametar koji predstavlja težine za svaki odbirak posebno, i on ima ulogu da uzme u obzir odbirke koji su bliži i-tom odbirku za koji računamo predikciju, a eleminiše ostale.

Da bi se izračunala srednja kvadratna greška, pronađene su predikcije za sve odbirke, pa je zatim vršeno njihovo poređenje sa originalnim vrednostima za y. Srednje kvadratna greška koja je izračunata iznosi:

3562.9832469556854

Polinomijalna regresija

Polinomijalna regresija je implementirana tako što su veštački izračunati feature-i za određeni stepen. Kada bi u skupu prediktora postojala dva obeležja x1 i x2, i kada bi željeni stepen polinoma bio 3, tada bi se na originalan skup dodala obeležja x1^3, x2^3, x1^2, x2^2, x1^2*x2, x1*x2^2, x1x2 bez ponavljanja obeležja.

Kada se proračuna ovaj dopunjen skup obeležja, moguće je pomoću linearnog modela dobiti znatno veću tačnost ako su u pitanju nelinearni podaci.

S obzirom da su ovde prisutni podaci uglavnom linearni, tačnost polinomijalne regresije je gora nego obične linearne. Kao primer uzet je polinom drugog stepena.

Ugrađena polinomijalna regresija: 4481.598729261616

Moja polinomijalna regresija: 4491.770245767902