**Regresia Liniară. Regresia Ridge**

1. **Regresia Liniară**

Dorim să găsim o funcție *g* astfel încât:

și care interpolează cel mai bine o mulțime de exemple

Pentru a găsi această funcție, vom minimiza valoarea funcției **M**ean **S**quared **E**rror pe mulțimea de antrenare.

1. **Regresia Ridge**

Regresia Ridge adaugă o nouă ”penalizare” funcției de cost, pe lângă faptul că diferența între etichetele *ground-truth* și etichetele *prezise* trebuie să fie minimă, dorim ca ponderile pe care le învățăm să fie mici. Pentru a forța ponderile să fie mici, vom adaugă la funcția de cost norma a ponderilor.

Parametrul controlează cât de mici să fie ponderile.

1. **Regresia Lasso**

Regresia Lasso adaugă norma a ponderilor la funcția de cost, creând o reprezentare *sparse* a ponderilor.

În acest laborator vom folosi modelele implementate în biblioteca Scikit-Learn.

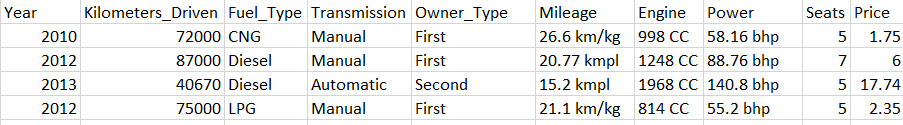
|  |
| --- |
| **from** sklearn.linear\_model **import** LinearRegression, Ridge, Lasso *# definirea modelelor* linear\_regression\_model = LinearRegression() ridge\_regression\_model = Ridge(alpha=1) lasso\_regression\_model = Lasso(alpha=1)  *# calcularea valorii MSE și MAE*  **from** sklearn.metrics **import** mean\_squared\_error, mean\_absolute\_error  mse\_value = mean\_squared\_error(y\_true, y\_pred)  mae\_value = mean\_absolute\_error(y\_true, y\_pred) |

***Car Price Prediction***

În continuare, vom lucra pe baza de date ***Car Price Prediction***  pentru a prezice prețul unei mașinii în funcție de caracteristicile ei.

Această bază de date este formată din 4879 exemple de antrenare. Neavând o mulțime separată de testare vom folosi tehnica de validare încrucișată (*cross-validation*) pentru a valida parametrii modelelor pe care le vom antrena.

În figura de mai jos, vedem 4 exemple din mulțime de antrenare.



După procesarea datelor (extragerea datelor din CVS și salvarea lor în format .npy) atributele au fost rearanjate în felul următor:

1. anul fabricației
2. numărul de kilometrii
3. mileage
4. motor
5. putere
6. numărul de locuri
7. numărul de proprietarii (valori între 1 și 4)

8-12. tipul de combustibil - fiind 5 tipuri de combustibil, acesta a fost recodat într-un one-hot vector de 5 componente.

13-14. tipul de transmisie - fiind 2 tipuri de transmisie, acesta a fost recodat într-un one-hot vector de 2 componente. 10 - „Manual”; 01 - ”Automatic”.

***Descărcați arhiva care conține datele de antrenare*** [***de aici.***](https://fmi-unibuc-ia.github.io/ia/Data/data_lab6.zip)

Codul următor ne ajută să citim datele de antrenare:

|  |
| --- |
| **import** numpy **as** np **from** sklearn.utils **import** shuffle  *# load training data* training\_data = np.load('data/training\_data.npy') prices = np.load('data/prices.npy') *# print the first 4 samples* print('The first 4 samples are:\n ', training\_data[:4]) print('The first 4 prices are:\n ', prices[:4]) *# shuffle*  training\_data, prices = shuffle(training\_data, prices, random\_state=0) |

**Exerciții**

1. Definiți o metodă care primește doi parametrii, datele de antrenare și cele de testare și returnează datele normalizate. Folosiți o metodă de normalizare ***corespunzătoare*** pentru setul de date ***Car Price Prediction***.
2. Folosind mulțimea de antrenare din setul de date ***Car Price Prediction***  antrenați un *model de regresie liniară* folosind validarea încrucișată cu 3 fold-uri. Calculați valoarea medie a funcțiilor *MSE* și *MAE*.

*Nu uitați să normalizați datele folosind metoda definită anterior.*

1. Folosind mulțimea de antrenare din setul de date ***Car Price Prediction***  antrenați un *model de regresie ridge* folosind validarea încrucișată cu 3 fold-uri. Calculați valoarea medie a funcțiilor *MSE* și *MAE*. Verificați care valoare a lui , obține o performanță mai bună.

*Nu uitați să normalizați datele folosind metoda definită anterior.*

1. Folosind cel mai performant *alpha* de la punctul anterior, antrenați un *model de regresie ridge* pe întreaga mulțime de antrenare, afișați coeficienți și bias-ul regresiei. Care este *cel mai semnificativ* atribut? Care este al doilea *cel mai semnificativ atribut*? Care este *cel mai puțin semnificativ* atribut?

*Nu uitați să normalizați datele folosind metoda definită anterior.*