Universitat Politècnica de Catalunya

Computació i Sistemes Intel·ligents

Yacht Hydrodynamics

Pràctica de Machine Learning

Carlota Catot Bragós

Francesc Reig Marsol

Quadrimestre de tardor 2019-2020

En aquest document s'exposa la resolució i les conclusions extretes de l'enunciat proposat per a la pràctica de *Machine Learning* corresponent a la tercera part de l'assignatura de CSI (Computació i Sistemes Intel·ligents).

1 Inspecció del *dataset*

Tal i com es pot observar en el títol del document, s'escull el *dataset* de **Yacht Hydrodynamics** per a aplicar models de **regressió** (aprenentatge automàtic supervisat).

Les dades s'han obtingut gràcies al **Dr. Roberto López** pertanyent al departament de Tecnologia i Transport Marítim de la Universitat Tècnica de Delf. El *dataset* conté 308 experiments a gran escala on s'inclouen 22 formes diferents de casc, derivades d'una forma relacionada amb el model "*Standfast 43*".

Segons l'enunciat del projecte, el model realitzat ha de predir l'última variable del *dataset* a partir de les 6 variables numèriques anteriors. Donat que les variables són coeficients, totes les variables són adimensionals.

Les variables relacionades amb els coeficients de geometria del casc i al nombre de Froude:

1. Posició longitudinal del centre de flotabilitat.
2. Coeficient prismàtic.
3. Relació longitud - desplaçament.
4. Relació mànega màxima - calat.
5. Relació longitud màxima - mànega màxima.
6. Nombre de Froude.

La variable que es vol predir és la resistència residual per unitat de pes de desplaçament:

7. Resistència residual per unitat de desplaçament.

A continuació, en la Figura 1 es mostren totes les dades del dataset[1].

[1]S'han hagut d'eliminar espais i caràcters *extra* del dataset que impedien la importació automàtica per a poder començar a tractar les dades.

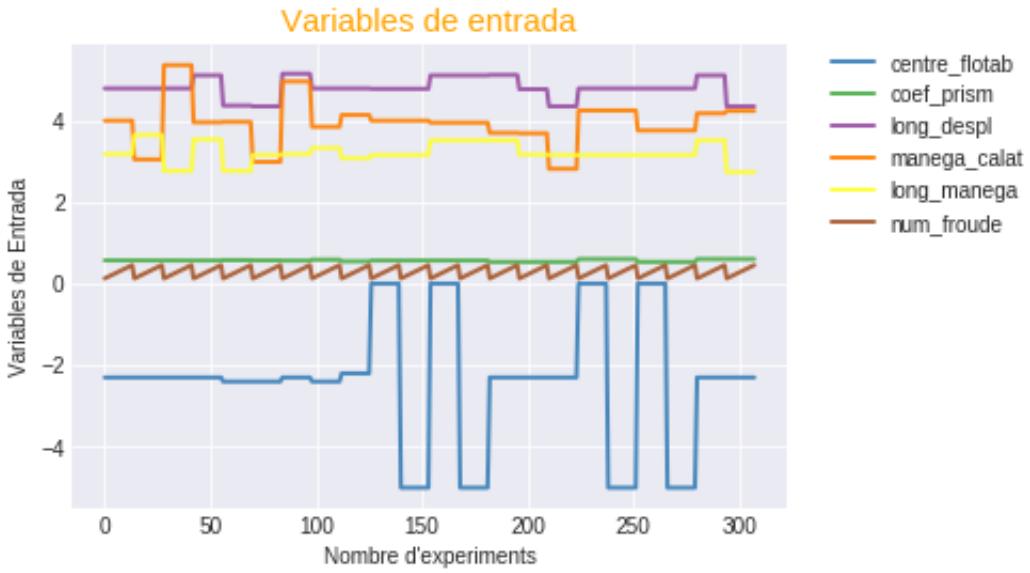

Figura 1: Representació de les dades del *dataset* Yacht Hydrodynamics. Font: elaboració pròpia.

1.1 Preprocessament de les dades

Donat que totes les dades que provenen del *dataset* són numèriques, no fa falta assignar-les amb valors numèrics (com passaria amb dades categòriques). No obstant, després d'observar la Figura 1, podem observar que no tots els atributs tenen el mateix rang. Per tant, per tal de donar el mateix pes a tots els atributs i, com a conseqüència independitzar-los de cada un d'ells, és necessari aplicar un escalat de totes les variables d'entrada.

Podem observar que les dades del *dataset* presenten valors negatius i positius, així doncs, per tal de mantenir la forma de la distribució de les dades, s'utilitza el mètode *MaxAbsScaler()* de la llibreria de *sklearn.preprocessing* el qual permet escalar i traslladar cada atribut d'entrada de forma que el valor absolut màxim sigui 1.0 (és a dir, totes les característiques del model passen a tenir valors del rang [-1, 1]).

A continuació, en la Figura 2 es mostra les dades després del preprocessament. Es pot observar sobretot com l'atribut *centre_flotab* manté la seva distribució en el *dataset*.

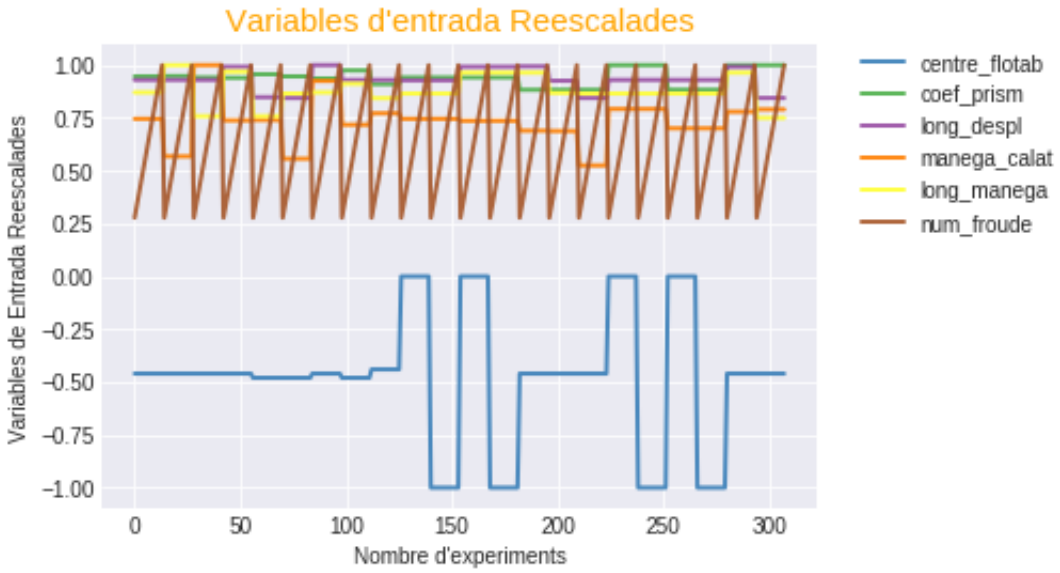

Figura 2: Representació de les dades reescalades del *dataset* Yacht Hydrodynamics. Font: elaboració pròpia.

2 Metodologia seguida

A continuació, es descriu la metodologia seguida per a la creació del model que ens permetrà esbrinar l'atribut de la resistència residual per unitat de desplaçament. Es descriuen els passos seguits per a la separació de les dades per a l'entrenament del model i per a la validació d'aquest i la selecció dels algorismes per a maximitzar els resultats.

2.1 Separació de les dades entre *train* i *test*

Per tal d'evitar l'*overfitting* a l'hora de realitzar l'entrenament del model escollit (secció 2.2.1), es separen les dades del *dataset* entre dades que s'utilitzaran per entrenar el model i dades que aquest no "haurà vist mai" que ens serviran per a comprovar l'eficàcia del model entrenat.

S'utilitza el mètode *train_test_split()* de la llibreria *sklearn.model_selection* per a dividir les dades. En aquest cas, s'han partit les dades en un 20% per a la validació del model (*test*) i el 80% restant per a l'entrenament d'aquest (*train*). A més, s'ha utilitzat una llavor per a generar el nombre aleatori i s'han remenat (*shuffle*) les dades abans de dividir-les.

2.2 Aplicació d'algorismes d'aprenentatge automàtic

En aquest apartat es descriu l'aplicació dels algorismes de *Machine Learning* que s'han discutit a les classes de teoria de l'assignatura com també alguns algorismes de regressió que presenta la llibreria de *scikit-learn*.

2.2.1 Validació creuada (*Hyperparameter Tunning*).

Per tal de mitigar l'*overfitting*, s'utilitza validació creuada o *Cross Validation* per tal que en cada iteració de l'entrenament del model utilitzi un conjunt de dades diferent i així la repetició de les dades no infereixin sobre el model.

S'ha utilitzat el mètode *Exhaustive Grid Search* (*GridSearchCV()*, en *scikit-learn*), el qual utilitza la "força bruta" per trobar la millor combinació de paràmetres (entrats a priori) que necessita el model escollit.

2.2.2 Algorismes de regressió lineal.

Els algorismes que s'han utilitzat per a l'entrenament del model són els següents:

1. Regressió Lineal.
2. Regressió de Ridge.
3. Regressió de Lasso.
4. Arbres de regressió..
5. Regressió *Random Forest*.
6. Regressió AdaBoost.

Finalment, de cada un dels algorismes executats per al model s'han obtingut les mètriques que ens dónen la informació següent:

- **R2**: coeficient de determinació que serveix per a avaluar si el mètode aplicat (a través de la comparació entre las prediccions i els valors reals) realitza una regressió correcta. Valors propers a l'1 indica que els valors predits pel model s'acosten als valors reals.
- **Mean Squared Error**: estimador que mesura la mitjana de les arrels dels errors o derivacions entre els resultats esperats i els que prediu el mètode de regressió aplicat. Per tant, el valor de l'error quadràtic mig ha de ser el mínim possible.
- **Root Mean Squared Error**: valor eficaç de la mètrica anterior que ens permet comparar com de diferent són, en valor mig, les prediccions respecte als valors originals (eliminant el soroll i la mida de les variables). Per tant, aquesta mètrica també ha de ser la mínima possible.
- **Mean Absolute Error**: estimador que permet obtenir la mitjana de l'error absolut. Per tant, com més petit és la mètrica, millor és el model utilitzat.
- **Median Absolute Error**: estimado ruq epermet obtenir la mediana de l'error absolut. Un valor petit indica que les dades predites s'allunyen prop de les originals i que, per tant, ens indica que el model s'acosta a la realitat.

3 Resultats

3.1 Mètriques dels models

Una vegada s'han realitzat tots els entrenaments pels algorismes enumerats en l'apartat 2.2.2 i havent realitzat la validació creuada de l'apartat 2.2.1, s'han obtingut els següents resultats de les mètriques de cada model (Taula 1, 2 i 3).

Taula 1: Mètriques obtingudes de la Regressió Lineal, Regressió de Ridge i Regressió de Lasso. Font: elaboració pròpia

Tipus Regressió	**Regressió Lineal**	**Regressió de Ridge**	**Regressió de Lasso**
Temps d'entrenament	1.02257	1.09331	0.17031
Puntuació GridSearchCV	0.54503	0.56771	-0.12526
Millor paràmetre GRCV	'copy_X': 'True', 'fit_intercept': 'True', 'normalize': 'True'	'alpha': 1.0	'alpha': 1
R2	0.54503	0.56771	-0.12526
MSE	0.01735	0.01649	0.04291
RMSE	0.13172	0.10286	0.20715
MAE	0.10906	0.12840	0.16795
MEDAE	0.10367	0.10199	0.160105
R2 Reescalat	0.54503	0.56771	-0.12526
MSE Reescalat	67.60264	64.23309	167.19850
RMSE Reescalat	8.22208	8.01456	12.93053
MAE Reescalat	6.80753	6.42052	10.48361
MEDAE Reescalat	6.47116	6.36631	9.99378

Taula 2: Mètriques obtingudes del model Arbre de Regressió, Regressió Random Forest i Regressió AdaBoost. Font: elaboració pròpia

Tipus Regressió	**Arbres de Regressió**	**Regressió Random Forest**	**Regressió AdaBoost**
Temps d'entrenament	**1.29804**	59.60951	22.56656
Puntuació GridSearchCV	**0.99523**	0.95591	0.98855
Millor paràmetre GRCV	**max. depth: 5**	'colsample_bytree': 0.7, 'gamma': 0.3, 'max_depth': 3, 'min_child_weight': 4, 'subsample': 0.9	'learning_rate': 1, 'loss': 'linear', 'n_estimators': 50
R2	**0.99523**	0.955912	0.98855
MSE	**0.00018**	0.00168	0.00044
RMSE	**0.01349**	0.04101	0.02090
MAE	**0.00678**	0.03166	0.01774
MEDAE	**0.00279**	0.03139	0.01951
R2 Reescalat	**0.99523**	0.95591	0.98855
MSE Reescalat	**0.709246**	6.55087	1.70191
RMSE Reescalat	**0.84217**	2.55947	1.30457
MAE Reescalat	**0.42306**	1.97596	1.30457
MEDAE Reescalat	**0.17426**	1.95957	1.21842

Es pot observar que a les taules també s'ha adjuntat les mètriques reescalades. Aquestes contenen els resultats de les mètriques amb les dades del *dataset* original, ja que d'aquesta manera es pot realitzar les comparacions sense la influència de l'escalat que s'ha realitzat en l'apartat 1.1, és a dir, les mètriques annexades a les distàncies relatives de la distribució de les dades (MSE, MEDAE, etc.) proporcionen una informació que s'acosta més a la realitat.

3.2 Representació dels resultats obtinguts

A continuació, es mostren les gràfiques obtingudes per a totes les mètriques sense reescalar i reescalades. Les figures mostren, per a tots els models utitlizats, la distribució dels valors reals de la resistència residual per unitat de desplaçament (sense reescalar i reescalat) respecte als valors predits per a cada un dels models.

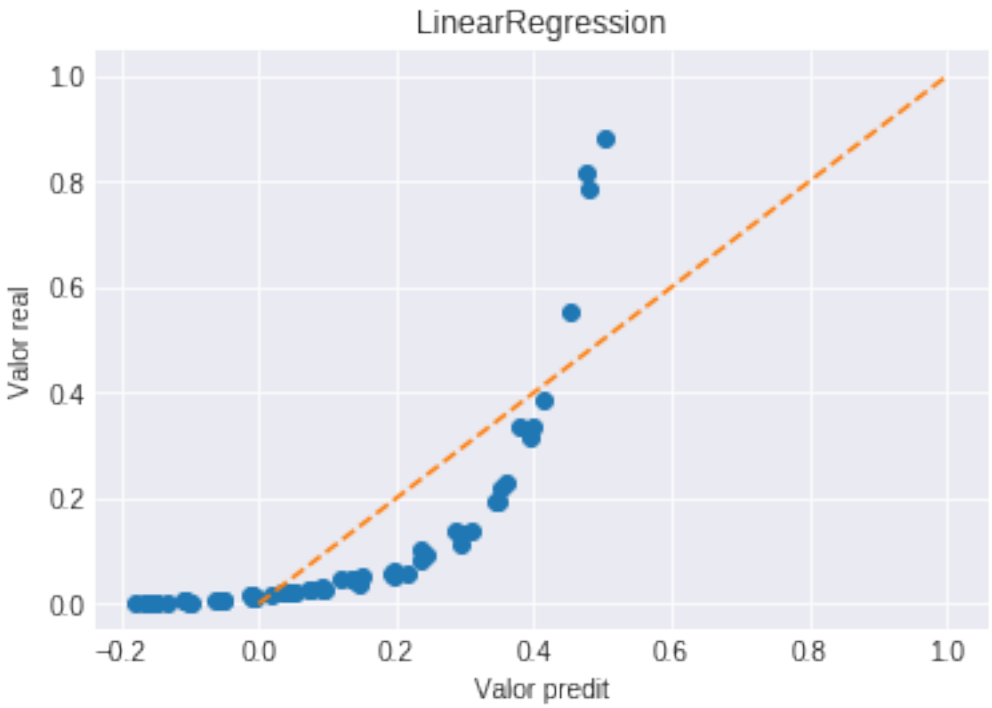

Figura 3: Valor real vs. Valor predit en el model de Regressió Lineal per a la resistència residual per unitat de desplaçament. Dades sense reescalar. Font: elaboració pròpia.

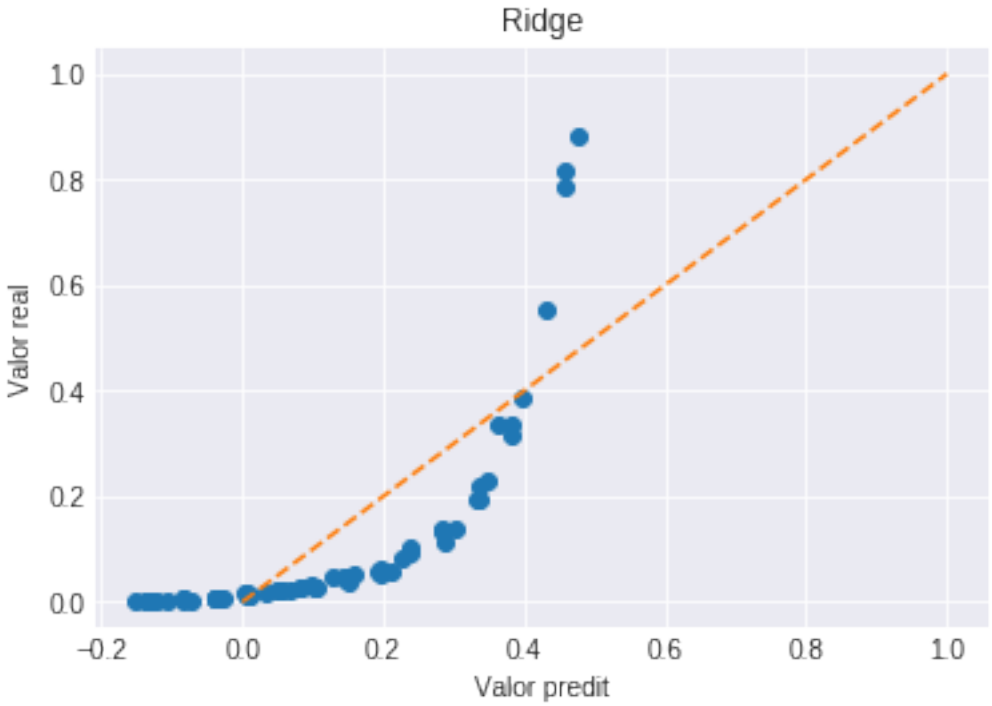

Figura 4: Valor real vs. Valor predit en el model de Regressió de Ridge per a la resistència residual per unitat de desplaçament. Dades sense reescalar. Font: elaboració pròpia.

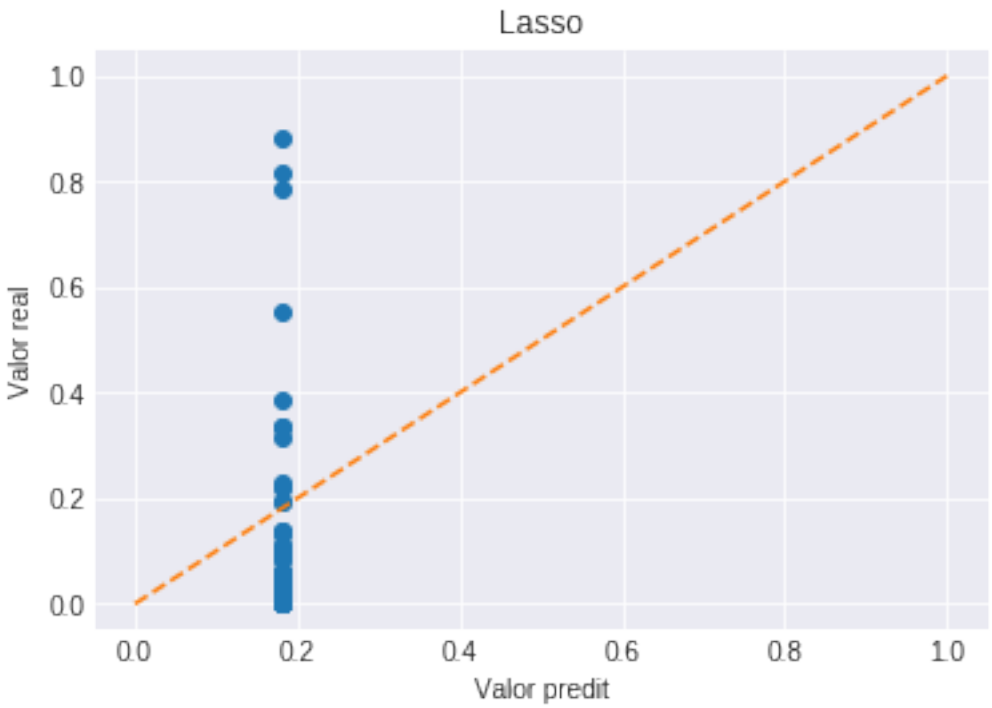

Figura 5: Valor real vs. Valor predit en el model de Regressió de Lasso per a la resistència residual per unitat de desplaçament. Dades sense reescalar. Font: elaboració pròpia.

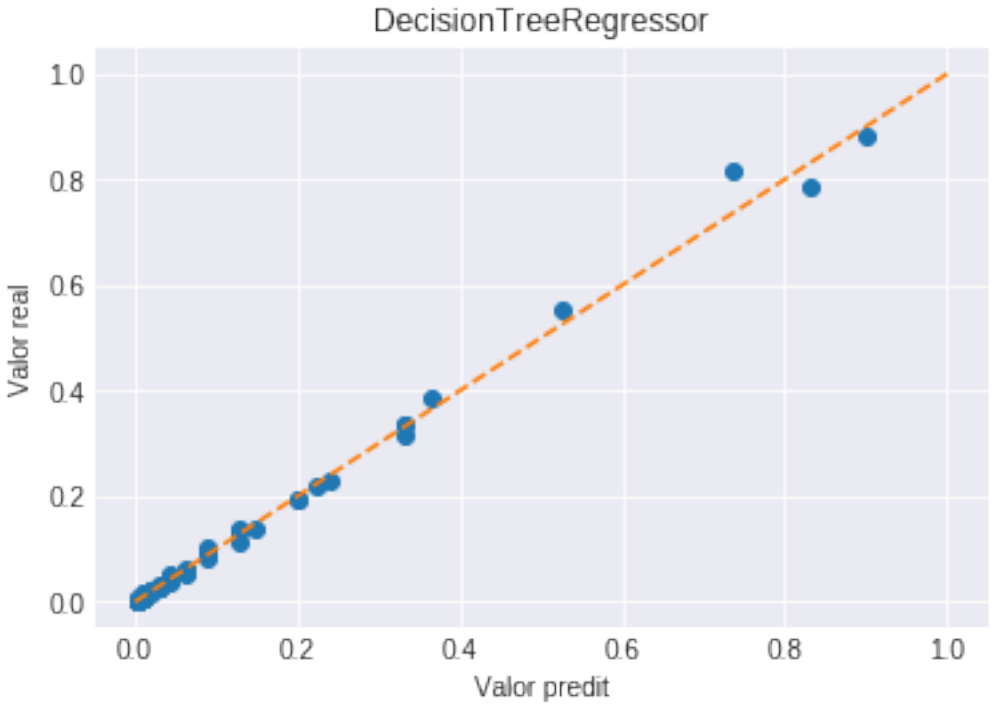

Figura 6: Valor real vs. Valor predit en el model d'Arbre de Regressió per a la resistència residual per unitat de desplaçament. Dades sense reescalar. Font: elaboració pròpia.

Figura 7: Valor real vs. Valor predit en el model de Regressió *Random Forest* per a la resistència residual per unitat de desplaçament. Dades sense reescalar. Font: elaboració pròpia.

Figura 8: Valor real vs. Valor predit en el model de Regressió Adaboost per a la resistència residual per unitat de desplaçament. Dades sense reescalar. Font: elaboració pròpia.

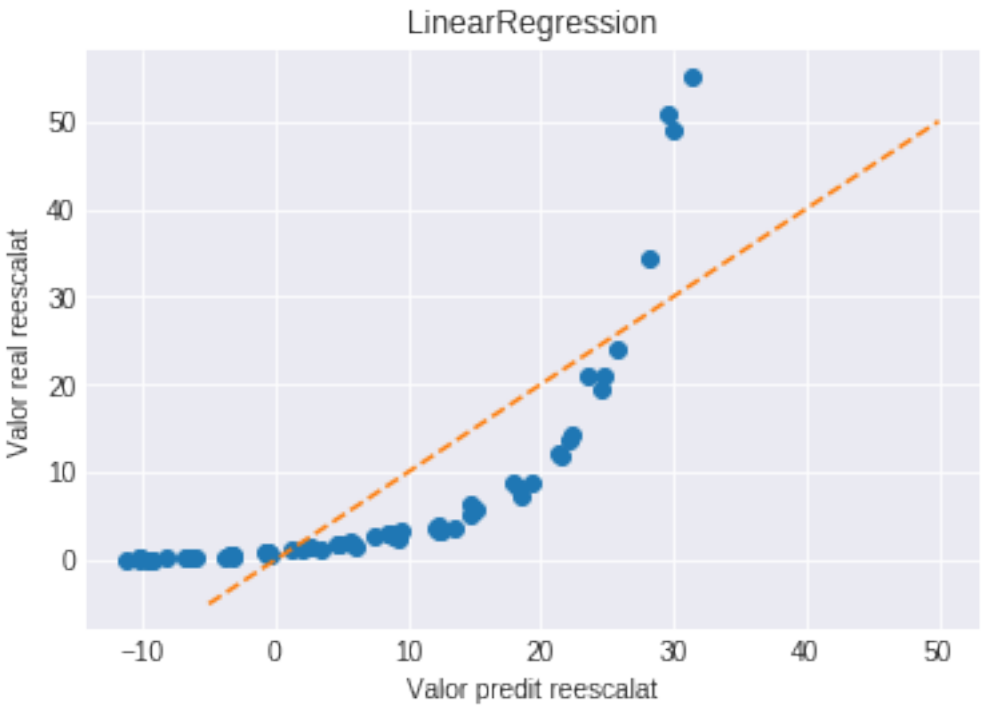

Figura 9: Valor real vs. Valor predit en el model de Regressió Lineal per a la resistència residual per unitat de desplaçament. Dades reescalades. Font: elaboració pròpia.

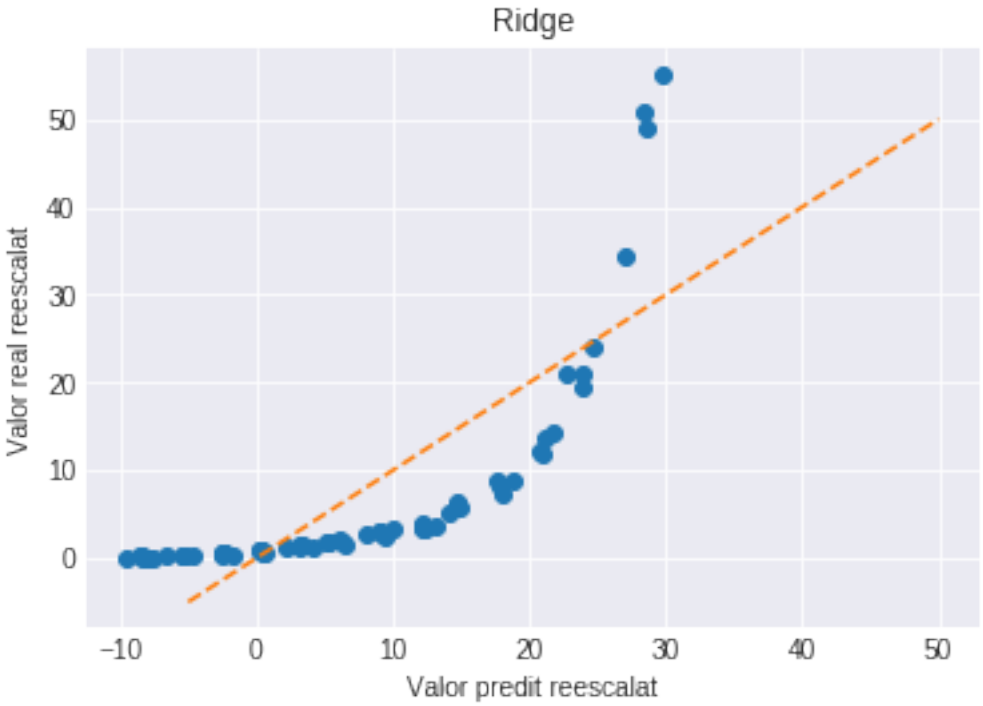

Figura 10: Valor real vs. Valor predit en el model de Regressió de Ridge per a la resistència residual per unitat de desplaçament. Dades reescalades. Font: elaboració pròpia.

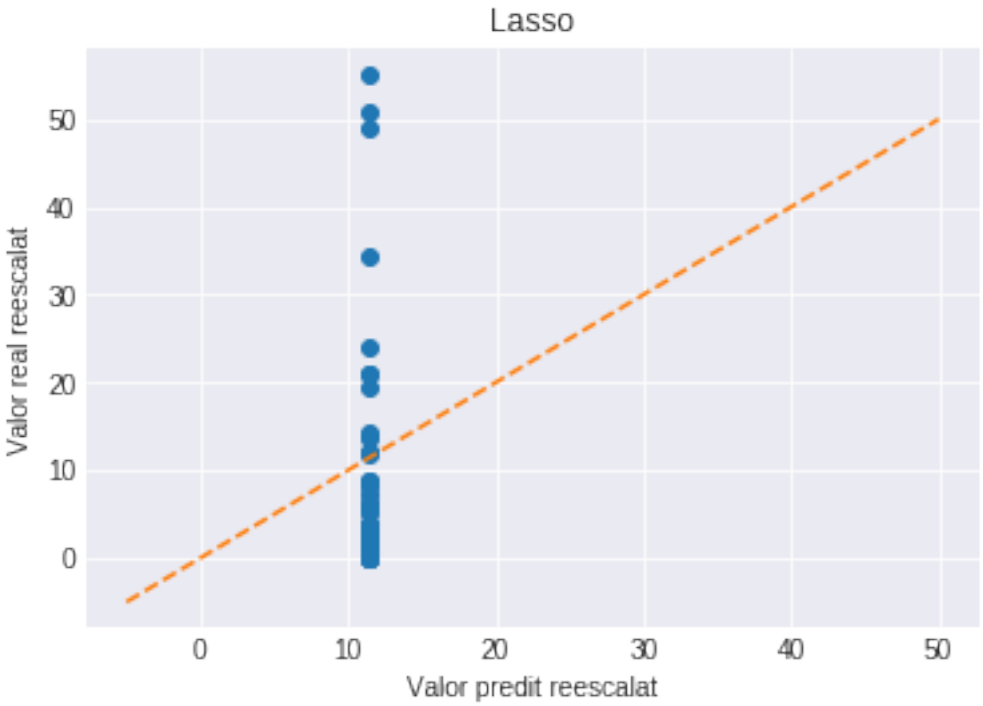

Figura 11: Valor real vs. Valor predit en el model de Regressió de Lasso per a la resistència residual per unitat de desplaçament. Dades reescalades. Font: elaboració pròpia.

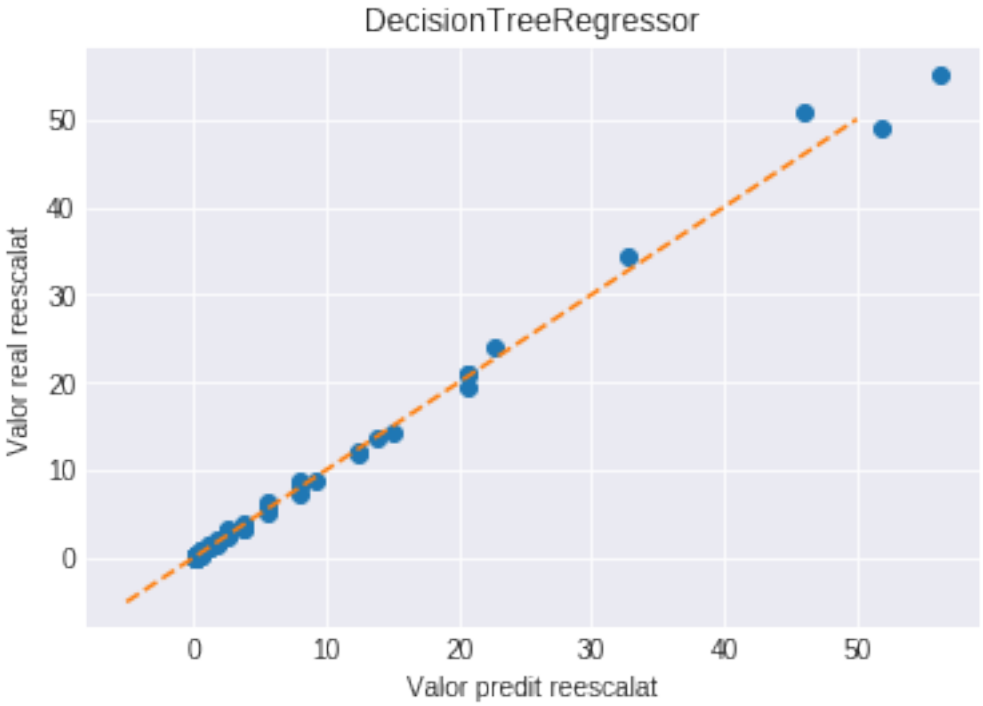

Figura 12: Valor real vs. Valor predit en el model d'Arbre de Regressió per a la resistència residual per unitat de desplaçament. Dades reescalades. Font: elaboració pròpia.

Figura 13: Valor real vs. Valor predit en el model de Regressió *Random Forest* per a la resistència residual per unitat de desplaçament. Dades reescalades. Font: elaboració pròpia.

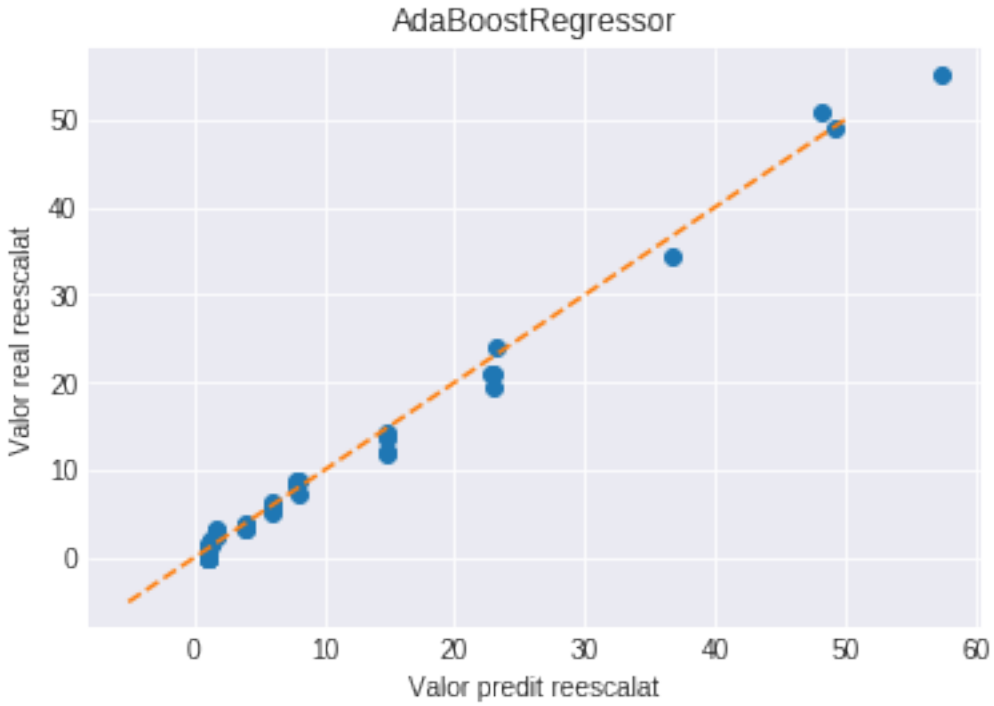

Figura 14: Valor real vs. Valor predit en el model de Regressió Adaboost per a la resistència residual per unitat de desplaçament. Dades reescalades. Font: elaboració pròpia.

3.3 Discussió de resultats

Tal i com es pot observar de les Figures 3 a 14, per a inspecció visual es pot preveure que els models que millor poden predir la resistència residual per unitat de desplaçament són l'**Arbre de Regressió**, seguit de la la Regressió *AdaBoost* i la Regressió *Random Forest*.

Podem observar també que si l'objectiu d'aquest projecte és trobar un model que ens permeti reduir l'error de generalització, observant la Taula 1 i 2 s'observa que efectivament la mètrica R2, que ens informa sobre la linealització que tenen les dades predites respecte a les reals, per al model d'Arbres de Regressió és el quin obté la millor puntuació (en el rang [0, 1] que pot assolir aquesta mètrica). A més a més, observant altres mètriques del model es pot observar com, per exemple l'error quadràtic mitjà també és mínim per aquest model.

Per tal d'evitar la influència que poden tenir les dades donat el seu escalat utilitzant l'escalador *MaxAbsScaler()*, les dades s'han reescalat una altra vegada als valors originals. Evidentment, el coeficient R2 es mantindrà, donat que la distribució de les dades no canvia, però sí que ho fan les altres mètriques, ja aquestes tenen en compte les distàncies relatives de les dades, i, com que el conjunt es reescala, les distàncies relatives també ho fan.

S'ha de tenir en compte que els models han estat entrenats amb els millors hiperparàmetres que el *GridSearchCV()* ha trobat, respecte dels paràmetres que s'ha "imposat". Per tant, hi ha hagut una influència externa que afecta a com el model ha estat entrenat i que no s'ha mesurat en aquest projecte.

De les mètriques i les gràfiques obtingudes en el subapartat anterior, s'observa que els models de Regressió Lineal, Regressió de Ridge i Regressió de Lasso no s'entrenen correctament, fins i tot amb els millors hiperparàmetres que s'han utilitzat per a aquest entrenament. A més, es pot observar com el millor hiperparàmetre α de Ridge i Lasso que ens dóna el model és 1, és a dir que el model no es regularitza i, per tant, no es disminueix la complexitat d'aquest per a l'entrenament. No obstant, podem observar com per a l'Arbre de Regressió el millor hiperparàmetre d'entrenament corresponent a la profunditat màxima de l'arbre és 5, enlloc de 11 (que correspon al màxim que s'ha imposat).

3.4 Reptes i Dificultats

Una vegada s'ha realitzat el projecte, es pot observar que contra tot pronòstic els mètodes que s'han introduït en les classes de teoria de l'assignatura no han proporcionat una sortida esperada. Això ha fet que es provessin altres models de *Machine Learning* que permetessin trobar una sortida esperada.

Una altre repte ha estat la comprensió de les dades numèriques i quina relació tenen totes les variables entre elles respecte a la seva distribució. Per tal d'evitar la deformació excessiva de les dades, s'ha investigat sobre quins tipus d'escaladors són necessaris per a quan es presenten moltes dades numèriques negatives. A més, s'han observat efectes contraproduents (com la intensificació del soroll de les variables) en els resultats finals que han fet reconduir el projecte des de l'inici (reescalat diferent).

4 Codi

En aquest projecte s'ha utilitzat *Python* per entrenar els models de *Machine Learning* mitjançant *Jupyter Notebook*. En el fitxer comprimit que s'envia juntament amb aquest informe, hi ha el fitxer *ProyectoML.ipynb* on es escriu pas per pas la obtenció dels resultats de l'apartat anterior.