CS-482/CS-682 Machine Learning

Regression and Classification Metrics

In this assignment you will demonstrate your knowledge of metrics for regression and classification. You must turn in a report providing answers for the sections below and also a python file containing code. As always place all your files into a folder with your lastnames# and zip that folder.

REGRESSION TASK: Use one the following datasets for regression from the UCI database

https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set

or another dataset of your choice.. With no missing values and not containing categorical variables.

a) **Load the Data:** Either write your own load function or use one the scikit-learn functions or pandas function to load the data and receive a class containing the numpy array of shape (n,p+1)

b) **Meet the Data Section:** Provide the following information about the data.

   a) Number of features (comes from the load function above)
   b) Names of the features (comes from load function above)
   c) Name of target (comes from load function above)
   d) Number of samples (comes from the load function above)
   e) Description of data (comes from load function above)
   f) First five rows of the data
   g) ***Correlation Studies:*** Present the correlation between each pair of data columns including target as follows:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	PRICE
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
TRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
PRICE	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

output of bostan.corr()

c) **Model Fitting and Parameter tuning:** Find the best parameter alpha: 0, 0.1, 1.0, 10.0, 20.0 50.0, 100.0 for both Ridge Regression and Lasso Regression with cross validation (5 fold). Use GridSearchCV from scikit-learn for this purpose. Provide the best parameter found for Ridge and Lasso Regression.

d) **Metrics:** Print the metrics R^2, RMSE (square root MSE). Use the best parameter found in the section above for Ridge and Lass regression

	R^2	RMSE
Lasso Regression		
Ridge Regression		
Linear Regression		

e) **Comparison of Different Models:** Provide the parameter values for Lasso, Ridge and Linear Regression as shown below, (except the diagram should include best alpha found for Lasso and Ridge unlike what is shown)

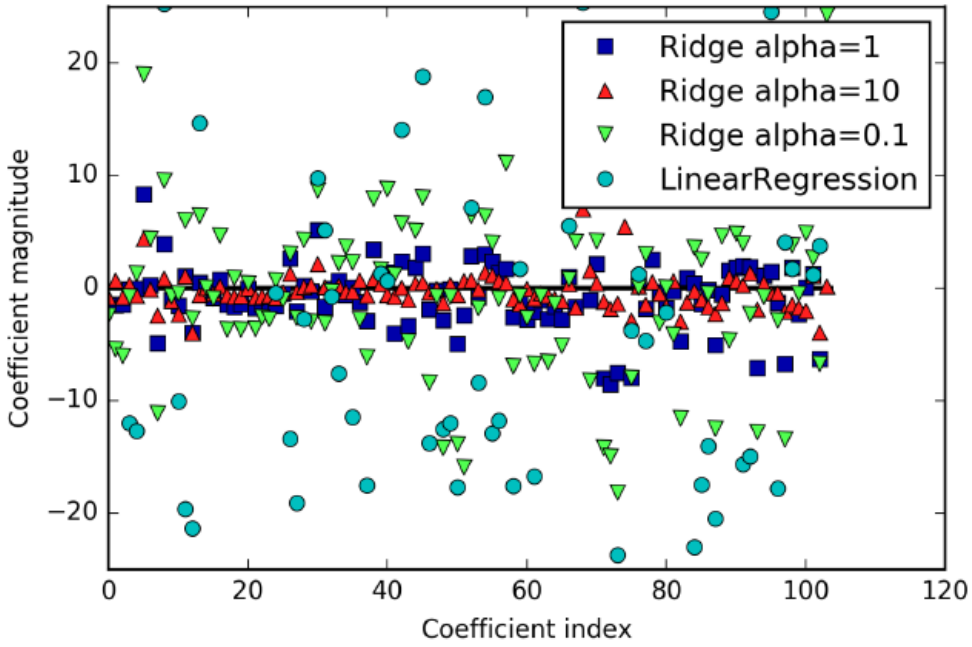


Figure 2-12. Comparing coefficient magnitudes for ridge regression with different values of alpha and linear regression

CLASSIFICATION TASK : Use one of the following datasets meant for binary classification.

https://archive.ics.uci.edu/ml/datasets/Haberman%27s+Survival

https://archive.ics.uci.edu/ml/datasets/banknote+authentication

or your own dataset which must be binary classification data with no missing values and no categorical data.

a) Either write your own load function or use one the scikit-learn functions to load the data and receive a class containing the numpy array of shape (n,p+1)
b) **Meet the Data Section:** Provide the following information about the data.

   a) Number of features (comes from the load function above)
   b) Names of the features (comes from load function above)
   c) Name of target (comes from load function above)

d) Number of samples (comes from the load function above)
e) Description of data (comes from load function above)
f) First five rows of the data
g) ***Correlation Studies:*** Present the correlation between each pair of data columns including target as follows:

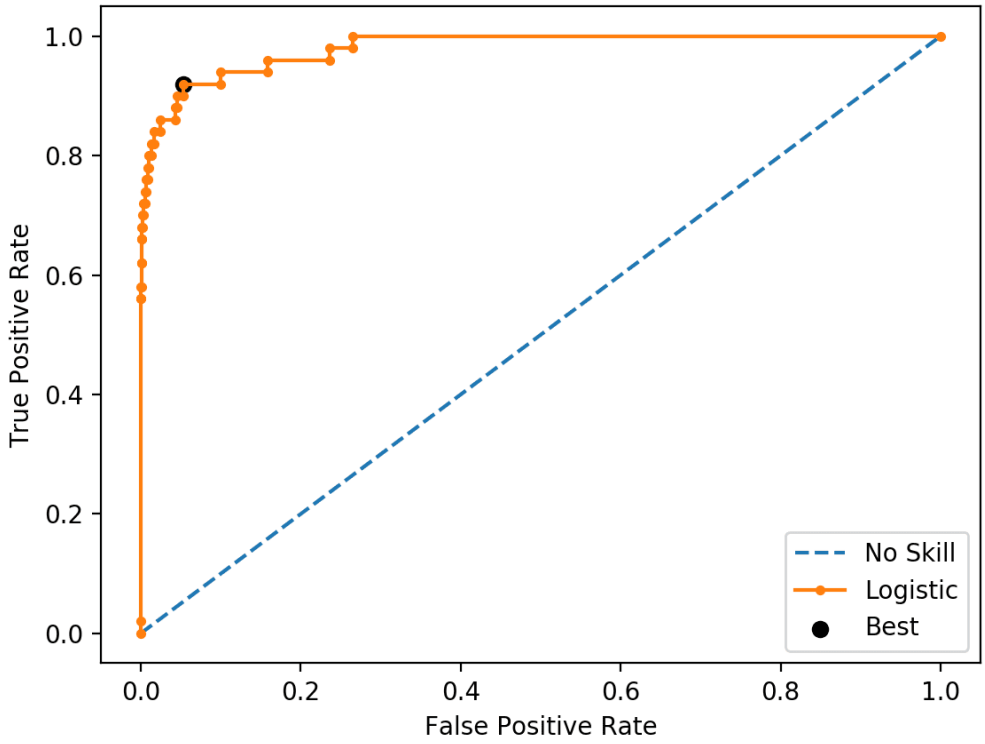
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	PRICE
CRIM	1.000000	-0.199458	0.404471	-0.055295	0.417521	-0.219940	0.350784	-0.377904	0.622029	0.579564	0.288250	-0.377365	0.452220	-0.385832
ZN	-0.199458	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948	-0.314563	-0.391679	0.175520	-0.412995	0.360445
INDUS	0.404471	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129	0.720760	0.383248	-0.356977	0.603800	-0.483725
CHAS	-0.055295	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368	-0.035587	-0.121515	0.048788	-0.053929	0.175260
NOX	0.417521	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441	0.668023	0.188933	-0.380051	0.590879	-0.427321
RM	-0.219940	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847	-0.292048	-0.355501	0.128069	-0.613808	0.695360
AGE	0.350784	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022	0.506456	0.261515	-0.273534	0.602339	-0.376955
DIS	-0.377904	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588	-0.534432	-0.232471	0.291512	-0.496996	0.249929
RAD	0.622029	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000	0.910228	0.464741	-0.444413	0.488676	-0.381626
TAX	0.579564	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228	1.000000	0.460853	-0.441808	0.543993	-0.468536
TRATIO	0.288250	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741	0.460853	1.000000	-0.177383	0.374044	-0.507787
B	-0.377365	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413	-0.441808	-0.177383	1.000000	-0.366087	0.333461
LSTAT	0.452220	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676	0.543993	0.374044	-0.366087	1.000000	-0.737663
PRICE	-0.385832	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626	-0.468536	-0.507787	0.333461	-0.737663	1.000000

output of bostan.corr()

c) **Model Fitting** Perform Logistic regression without cross validation. Split the data into 80% training and 20% test. Note that Logistic Regression on scikit-learn by default uses L2 penalty of 1. Each time use the same test set for comparison. State the accuracy.

d) **Metrics:** Logistic Regression can predict class labels or class probabilities (Continuous valued prediction) Using class label predictions, compute
   a) Precision,
   b) Recall,
   c) Sensitivity
   d) Accuracy

e) **Metrics:** Create an ROC curve and find the best threshold as shown below


True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
No Skill
Logistic
Best