CS412 Machine Learning - Homework 4 Linear Regression and Evaluation Metrics

Deadline: 30 April 2020, 23:55
Late submission: till 2 May 2020, 23:55
(-10pts penalty for **each** late submission day)

Submission
For your notebook results, make sure to run all of the cells and the output results are there.
Please submit your homework as follows:

- Download the .ipynb and the .py file and upload both of them to sucourse.
- Submit also a single pdf document by solving questions on the sheet.
- Link to your Colab notebook (obtained via the share link in Colab) in the sheet:

  https://colab.research.google.com/drive/1dALzOhHvNA3Upjqu503bLpTxIkeGI-ek

Objective
The topic of this homework assignment is supervised learning. The first half is concerned with linear regression, and the second half, performance measure on classification tasks.

Startup Code
https://colab.research.google.com/drive/1W80EpGJYudkQ7Sz2pbAHffvt9bo_ITHH
To start working for your homework, take a copy of this folder to your own google drive.

Software: You may find the necessary function references here:

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.RidgeCV.html

Question 1: 75 pts - Predict the price of houses.

Dataset Description

https://raw.githubusercontent.com/OpenClassrooms-Student-Center/Evaluate-Improve-Models/master/house_prices.csv

In this dataset, there are 2930 observations with 305 explanatory variables describing (almost) every aspect of residential homes.

a) **Find the correlation between garage area and sale price by applying linear regression. Print the bias and slope. Print the train and test R2. Plot the test set with a scatter plot and add the linear regression model line.**

```
[ ]  1 from sklearn.metrics import r2_score
     2
     3
     4 pred_y = regressor.predict(train_x)
     5 # code comes here
     6 print("Train:", r2_score(train_y,pred_y))
     7
     8 pred_y = regressor.predict(test_x)
     9 # code comes here
    10 print("Test:", r2_score(test_y,pred_y))
```

```
Train: 0.39923858732928674
Test: 0.4504070679149049
```

```
[ ]  1 print('Regressor coeffient or slope:',regressor.coef_[0][0])
     2 print('Interception point with axis:',regressor.intercept_[0])
```

```
Regressor coeffient or slope: 233.2031085567466
Interception point with axis: 70159.21749224591
```

Positive correlation and meaningful but low R2 scores.

b) **Apply multiple linear regression by taking all input features. Print the train and test R2.**

```
[ ]  1 pred_y = regressor.predict(train_x)
     2 # code comes here
     3 print("Train:", r2_score(train_y,pred_y))
     4
     5 pred_y = regressor.predict(test_x)
     6 # code comes here
     7 print("Test:", r2_score(test_y,pred_y))
```

```
Train: 0.9430380214000639
Test: -1.9758405174869437e+17
```

c) **Comment on part a and b results. Why R2 is low in part a? Why test R2 is low although train R2 is quite high in part b?**

In part a, there is not enough correlation between the Garage Area and Price Sales to predict Price Sales accurately, so the model is too simple thus the low R2 score. In part b, model is overfitting because of too much features, so the model is too complex for the model the learn properly, thus model gives high train R2 score(memorized) and low test R2 score.

d) **Apply ridge regression with cross-validation by taking all input features. Print optimal alpha. Print also the train and test R2.**

```
[ ]  1 pred_y = model_rcv.predict(train_x)
     2 # code comes here
     3 print("Train:", r2_score(train_y,pred_y))
     4
     5 pred_y = model_rcv.predict(test_x)
     6 # code comes here
     7 print("Test:", r2_score(test_y,pred_y))
```

```
Train: 0.9211536946303333
Test: 0.841271168982664
```

Print the best alpha.

```
[ ]  1 print("Alpha:", model_rcv.alpha_)
```

```
Alpha: 5.0
```

e) **Discuss on regularization. What is ridge regression? When do we use it? And what is the effect on features?**

Regularization means making things more regular or acceptable, in this case regularization works with adding a penalty term as in the form of a cost function in the linear regression model to make the model work better by avoiding too extreme coefficients. Ridge Regression is a type of regularization (also known as L_2 regularized regression). We use regularization in the case of overfitting models which have low biases but very high variances. Ridge Regression basically aims to decrease the variance by increasing the bias by a little bit and thus avoiding overfitting in the models. Also, the Ridge Regression can be cross validated to hyper-tune the alpha parameter and have the best result possible.

f) **Print regression coefficients for multiple linear regression and ridge regression. Comment on the change of feature weights. What is the effect of ridge regression on feature weights?**
The coefficients (feature weights) in the ridge regression are closer to 0 than the coefficients in the multiple linear regression which makes sense because ridge regression aims to make the feature weights less extreme.

Question 2: 25 pts - Evaluation metrics.

a) **15 pts - Provide the Confusion Matrix, Accuracy, Error, Precision, Recall, and F1-Score for the fruit classification problem. The output of test data classification results is given in the following table.**
Use both macro and micro averaging methods.

mass	width	height	color_score	class	prediction
154	7.1	7.5	0.78	orange	lemon
180	7.6	8.2	0.79	orange	lemon
154	7.2	7.2	0.82	orange	apple
160	7.4	8.1	0.80	orange	orange
164	7.5	8.1	0.81	orange	apple
152	6.5	8.5	0.72	lemon	lemon

118	6.1	8.1	0.70	lemon	apple
166	6.9	7.3	0.93	apple	apple
172	7.1	7.6	0.92	apple	apple

Confusion Matrix

		True Class		
		Orange	**Lemon**	**Apple**
	Orange	**1**	**0**	**0**
	Lemon	**2**	**1**	**0**
	Apple	**2**	**1**	**2**

Accuracy = (1+1+2)/9 = 0.44
Error = (2+2+1)/9 = 0.56

Macro – Averaging
Precision = (Precision (Orange) + Precision (Lemon) + Precision (Apple)) / 3
Precision = (1/1 + 1/3 + 2/5) / 3
Precision = (1 + 0.33 + 0.4) / 3 = 0.58

Recall = (Recall (Orange) + Recall (Lemon) + Recall (Apple)) / 3
Recall = (1/5 + 1/2 + 2/2) / 3
Recall = (0.2 + 0.5 + 1) / 3 = 0.57

F1 - Score = (2*Precision*Recall) / (Precision + Recall)
F1 – Score = (2*0.58*0.57) / 3 = 0.57

Micro – Averaging

Orange	True Yes	True No	Lemon	True Yes	True No	Apple	True Yes	True No
System Yes	1	0	System Yes	1	2	System Yes	2	3
System No	4	4	System No	1	5	System No	0	4

Pooled	True Yes	True No
System Yes	4	5
System No	5	13

Precision = 4/9 = 0.44
Recall = 4/9 = 0.44
F1-Score = 0.44

b) 10 pts - The table shows 18 data and the score assigned to each by a classifier. It is a binary classification problem. The active/decoy column shows the ground truth labels. Plot the corresponding ROC curve.

id	score	active/decoy	id	score	active/decoy
O	0.03	a	L	0.48	a
J	0.08	a	K	0.56	d
D	0.10	d	P	0.65	d
A	0.11	a	Q	0.71	d
I	0.22	d	C	0.72	d
G	0.32	a	N	0.73	a
B	0.35	a	H	0.80	d
M	0.42	d	R	0.82	d
F	0.44	d	E	0.99	d

Took decoy as positive. (True Positive Rate = TPR, False Positive Rate = FPR)

Score = 0.99
TPR = 1/11
FPR = 0/7

Score = 0.82
TPR = 2/11
FPR = 0/7

Score = 0.80
TPR = 3/11
FPR = 0/7

Score = 0.73
TPR = 3/11
FPR = 1/7

Score = 0.72
TPR = 4/11
FPR = 1/7

Score = 0.71
TPR = 5/11
FPR = 1/7

Score = 0.65
TPR = 6/11
FPR = 1/7

Score = 0.56
TPR = 7/11
FPR = 1/7

Score = 0.48
TPR = 7/11
FPR = 2/7

Score = 0.44
TPR = 8/11
FPR = 2/7

Score = 0.42
TPR = 9/11
FPR = 2/7

Score = 0.35
TPR = 9/11
FPR = 3/7

Score = 0.32
TPR = 9/11
FPR = 4/7

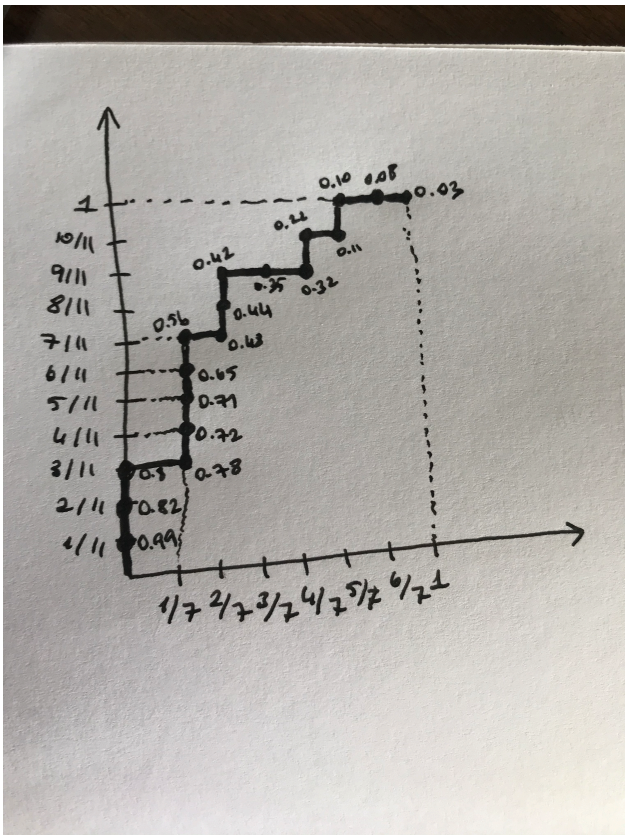
Score = 0.22
TPR = 10/11
FPR = 4/7

Score = 0.11
TPR = 10/11
FPR = 5/7

Score = 0.10
TPR = 11/11
FPR = 5/7

Score = 0.08
TPR = 11/11
FPR = 6/7

Score = 0.03
TPR = 11/11
FPR = 7/7


1
10/11
9/11
8/11
7/11
6/11
5/11
4/11
3/11
2/11
1/11
0.10
0.08
0.03
0.22
0.11
0.42
0.35
0.32
0.44
0.56
0.43
0.65
0.71
0.72
0.78
0.82
0.99
1/7
2/7
3/7
4/7
5/7
6/7
1