CS5014 P1: Regression Model

180007044
University of St Andrews
St Andrews, the United Kingdom
hs99@st-andrews.ac.uk

LOAD DATA

Since the provided dataset is saved in a CSV file, the first task is to load the data into session.

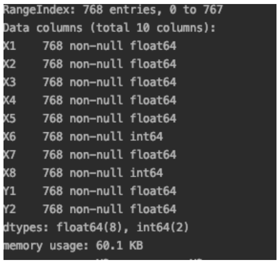
In this practical, Pandas is applied to load the data, since there are some other useful functions provided by this library. The data are loaded as Data Frame in this phase.

CLEAN DATA

There are some general steps to clean data, such as elimination missing values, tidying data and transforming data. In this practical, it could easily be noticed that the provided dataset is tidied, which means each column represents a separate variable, while each row represents an individual observation. [1]

Information of the Dataset

With the help of function *info (),* some information of the dataset could be explored.



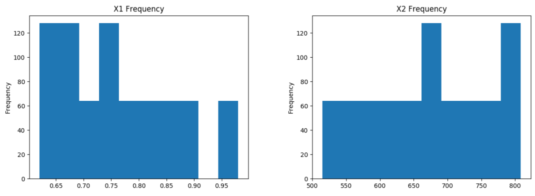
```
RangeIndex: 768 entries, 0 to 767
Data columns (total 10 columns):
X1    768 non-null float64
X2    768 non-null float64
X3    768 non-null float64
X4    768 non-null float64
X5    768 non-null float64
X6    768 non-null int64
X7    768 non-null float64
X8    768 non-null int64
Y1    768 non-null float64
Y2    768 non-null float64
dtypes: float64(8), int64(2)
memory usage: 60.1 KB
```

Figure 1. Information of the dataset

According to Fig. 1, all features are numerical and there are no missing values, hence it is not necessary to eliminate missing values and transform data types.

Explore Outliers

Histogram could be a virtue choice to explore outliers since it is objective and intuitive. [2] Therefore histograms are applied to find outliers for 8 input features.



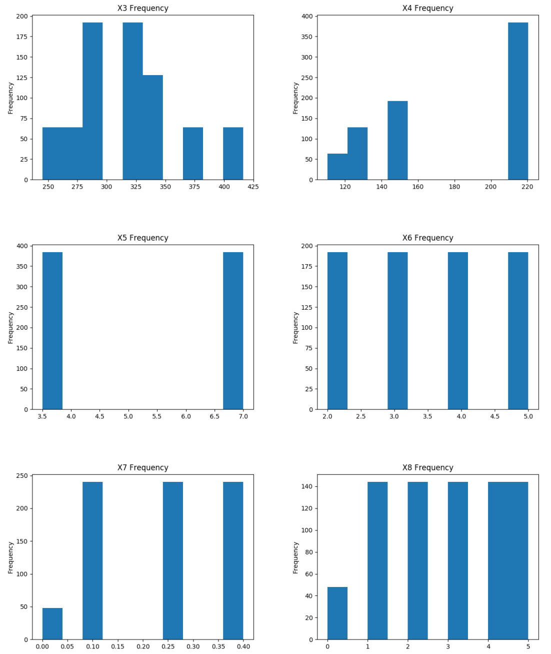


Figure 2. Density histograms of X1-X8

This group of density histograms of input variables could draw a conclusion that there are no outliers in these variables. Besides, almost all these features have limited number of possible values.

ANALYSE DATA

Before building a model, the data need to be analyzed in a statistical way.

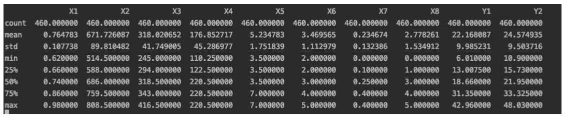
Split Dataset

To evaluate and report the performance of the model later, the dataset needs to be randomly split in to a training set, a validation set and a test set. The ratio is set to 3: 1: 1. Since all output variables are continuous, it is not necessary to use complex sampling methods such as stratified sampling.

All following analysis is based on the training set.

Basic Statistical Analysis

Using the function *describe (),* some basic statistical features of variables are demonstrated.



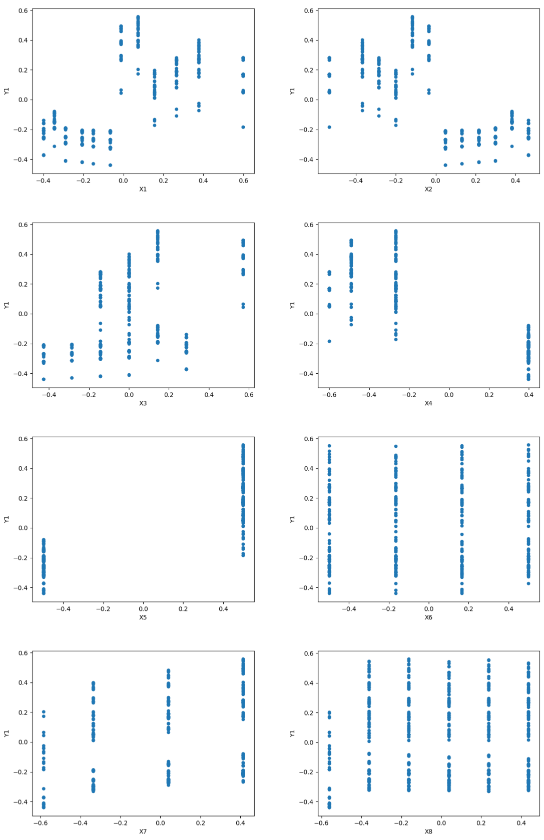
```
               X1          X2          X3          X4          X5          X6          X7          X8          Y1          Y2
count  460.000000  460.000000  460.000000  460.000000  460.000000  460.000000  460.000000  460.000000  460.000000  460.000000
mean     0.764783  671.726087  318.020652  176.852717    5.234783    3.469565    0.234674    2.778261   22.168087   24.574935
std      0.107738   89.810482   41.749005   45.286977    1.751839    1.112979    0.132386    1.534912    9.985231    9.503716
min      0.620000  514.500000  245.000000  110.250000    3.500000    2.000000    0.000000    0.000000    6.010000   10.900000
25%      0.660000  588.000000  294.000000  122.500000    3.500000    2.000000    0.100000    1.000000   13.007500   15.730000
50%      0.740000  686.000000  318.500000  220.500000    3.500000    3.000000    0.250000    3.000000   18.660000   21.950000
75%      0.860000  759.500000  343.000000  220.500000    7.000000    4.000000    0.400000    4.000000   31.350000   33.325000
max      0.980000  808.500000  416.500000  220.500000    7.000000    5.000000    0.400000    5.000000   42.960000   48.030000
```

Figure 3. Description of the dataset

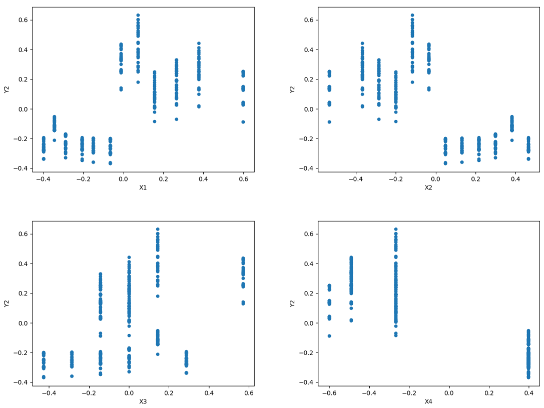
It could be seen that all input variables have different ranges, hence feature scaling is necessary. In this practical, since all features have definitive limitations and well-spaced values, they are normalized by their means to keep their value in the range -1 to 1, which could also decrease their standard errors.

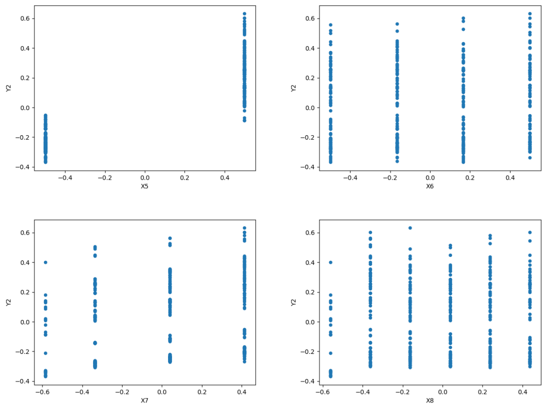
Data Plot

To explore relationships between input variables and output variables more intuitively, 16 scatter plots are drawn and divided into 2 groups by 2 output variables.



(a)





(b)

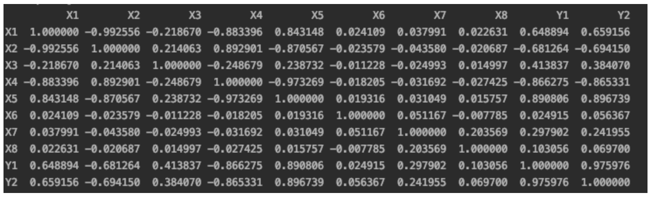
Figure 4. Scatter plot of input variables with Y1 (a) and Y2 (b)

As it is shown in these scatter plots, it could assume that some input variables such as X1, X2, X3, X4 and X5 may pose large influence on both two output variables, while X6, X7 and X8 may not extremely related to them. In next part, this assumption could be justified.

DATA PREPARATION

Feature Selection

In this practical, the features used to build models is selected mainly according to correlation coefficient between input variables and output variables. The threshold is set to 0.4 in this practical.



```
          X1        X2        X3        X4        X5        X6        X7        X8        Y1        Y2
X1  1.000000 -0.992556 -0.218670 -0.883396  0.843148  0.024109  0.037991  0.022631  0.648894  0.659156
X2 -0.992556  1.000000  0.214063  0.892901 -0.870567 -0.023579 -0.043580 -0.020687 -0.681264 -0.694150
X3 -0.218670  0.214063  1.000000 -0.248679  0.238732 -0.011228 -0.024993  0.014997  0.413837  0.384070
X4 -0.883396  0.892901 -0.248679  1.000000 -0.973269 -0.018205 -0.031692 -0.027425 -0.866275 -0.865331
X5  0.843148 -0.870567  0.238732 -0.973269  1.000000  0.019316  0.031049  0.015757  0.890806  0.896739
X6  0.024109 -0.023579 -0.011228 -0.018205  0.019316  1.000000  0.051167 -0.007785  0.024915  0.056367
X7  0.037991 -0.043580 -0.024993 -0.031692  0.031049  0.051167  1.000000  0.203569  0.297902  0.241955
X8  0.022631 -0.020687  0.014997 -0.027425  0.015757 -0.007785  0.203569  1.000000  0.103056  0.069700
Y1  0.648894 -0.681264  0.413837 -0.866275  0.890806  0.024915  0.297902  0.103056  1.000000  0.975976
Y2  0.659156 -0.694150  0.384070 -0.865331  0.896739  0.056367  0.241955  0.069700  0.975976  1.000000
```

Figure 5. Correlation coefficient within the dataset

Figure shows that X1, X2, X3, X4 and X5 are chosen to be used to predict Y1, while feature subset chosen to be used to predict Y2 contain X1, X2, X4 and X5, which is basically same to the assumption mentioned in last part.

REGRESSION MODEL

Select Model

For datasets who have multiple input variables, multivariate linear regression model could be an appropriate kind of linear model to predict a numerical output variable.

Therefore, to predict value of Y1 and Y2, two multivariate linear regression models would be built respectively.

The model for predicting Y1:

$$y_1 = \theta_o x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$$

$$y_1 = Y1, x_0 = 1, x_1 = X1, x_2 = X2, x_3 = X3, x_4 = X4, x_5 = X5$$

The model for predicting Y2:

$$y_2 = \theta_o x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$y_2 = Y2, x_0 = 1, x_1 = X1, x_2 = X2, x_3 = X4, x_4 = X5$$

Loss Function

In this practical, the loss function is defined as:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(\theta^T x^{(i)} - y^{(i)})^2$$

As two models are both linear regression models, they could share the same loss function.

Training

Having 5 and 4 features respectively, the models would be trained by normal equation, since it may have better efficiency and accuracy.

Regularization

To prevent that the models overfits the dataset, it is necessary to add regularization into the training process, which means the normal equation is changed to:

$$\theta = (X^T X + \lambda m)^{-1} X^T y$$

For the model predicting Y1, m is a 6x6 matrix in which only values on the leading diagonal besides the up-left corner are 1, while for that predicting Y2, m is a 5x5 matrix in the similar format.

The most work in this part is to find the most appropriate value for λ. The specific strategy would be introduced in optimization.

Initial Model

Based on above steps, there would be a pair of models successfully trained. In this phase, the value of lambda is initially set to 0.

Y1 model:

$$y_1 = 2.41 * 10^{-4} - 0.845x_1 - 0.884x_2 + 0.282x_3 - 0.0779x_4 + 0.342x_5$$

$$y_1 = Y1, x_0 = 1, x_1 = X1, x_2 = X2, x_3 = X3, x_4 = X4, x_5 = X5$$

Y2 model:

$$y_2 = 3.27x_0 * 10^{-3} - 0.738x_1 - 0.398x_2 - 0.297x_3 + 0.382x_4$$

$$y_2 = Y2, x_0 = 1, x_1 = X1, x_2 = X2, x_3 = X4, x_4 = X5$$

Evaluation

In this practical, the model is evaluated by the loss function and the coefficient of determination (R^2) of models. The dataset used for evaluation is the validation set split in preparation.

For the initial models, the loss and the coefficients of determination are listed in Table 1.

Table 1. Evaluation of the initial models

Predicted Variable	Loss	Coefficient of Determination
Y1	0.125757701	-2.489586026
Y2	0.020949785	0.338772124

The coefficients of determination show that the performance of models is extremely awful, hence it is highly necessary to optimize the models.

Optimization

To optimize the models, there are two strategies applied in this practical, one of which is to change to features, another one is to change the value of lambda.

Reselect features

In current models, there are 5 and 4 features involved respectively. Firstly, the rest features would be added into the model successively. If a model performs better after involving a certain feature, this feature would be kept in this model, otherwise it would be dropped again. Then, the features involved in the initial models would be similarly reassessed in this way. It could be regarded as a process combining both forward and backward stepwise selection.

As the result, X3 and X4 are dropped from the Y1 model, while the Y2 model is not changed. Following are models optimized after this phase:

Y1 model:

$$y_1 = -7.30 * 10^{-4} - 0.107x_1 + 0.231x_2 + 0.656x_4$$

$$y_1 = Y1, x_0 = 1, x_1 = X1, x_2 = X2,, x_4 = X5$$

Y2 model:

$$y_2 = 3.27x_0 * 10^{-3} - 0.738x_1 - 0.398x_2 - 0.297x_3 + 0.382x_4$$

$$y_2 = Y2, x_0 = 1, x_1 = X1, x_2 = X2, x_3 = X4, x_4 = X5$$

The performance of this pair of models are shown in Table 2.

Table 2. Evaluation of the models after reselecting features

Predicted Variable	Loss	Coefficient of Determination
Y1	0.013092815	0.636694180
Y2	0.020949785	0.338772124

It could be found that ahere is an obvious enhancement in performance of Y1 model.

Find suitable lambda

In models produced by above steps, the value of lambda is 0. To optimize the models, the value of lambda would change from 0 to 1, and based on its performance, continue to enlarge to 10 or shrink to 0.1. Then 100 or 0.01, 1000 or 0. 001....The performance of each step is listed in Table 3.

Table 3. Performance of models for different values of lambda

Value of lambda	Loss of Y1 model	R^2 of Y2 model	Loss of Y2 model	R^2 of Y1 model
0	0.013092	0.636694	0.020949	0.338772
1	180.1784	-4998.68	145.2223	-4582.58
0.1	1.958235	-53.3380	1.494741	-46.1777
0.01	0.046893	-0.30121	0.037865	-0.19512
0.001	0.014865	0.587499	0.021336	0.326559
0.0001	0.013254	0.632220	0.020975	0.337962
$1 * 10^{-5}$	0.013108	0.636251	0.020952	0.338695
$1 * 10^{-6}$	0.013094	0.636649	0.020950	0.338764
$1 * 10^{-7}$	0.013092	0.636689	0.020949	0.338771
$1 * 10^{-8}$	0.013092	0.636693	0.020949	0.338772
$1 * 10^{-9}$	0.013092	0.636694	0.020949	0.338772

As shown in Table 3, the value of lambda shrinks to an extremely tiny value, but the models do not perform better. It may be meaningless to continue this work; hence the lambda is still 0 finally.

Hence the final models are:

Y1 model:

$$y_1 = -7.30 * 10^{-4} - 0.107x_1 + 0.231x_2 + 0.656x_4$$

$$y_1 = Y1, x_0 = 1, x_1 = X1, x_2 = X2, , x_4 = X5$$

Y2 model:

$$y_2 = 3.27x_0 * 10^{-3} - 0.738x_1 - 0.398x_2 - 0.297x_3 + 0.382x_4$$

$$y_2 = Y2, x_0 = 1, x_1 = X1, x_2 = X2, x_3 = X4, x_4 = X5$$

Report Performance of the Models

Finally, using the test set to fit the models, the performance of the models is shown in Table 4.

Table 4. Final test result

Predicted Variable	Loss	Coefficient of Determination
Y1	0.016193362	0.59653274
Y2	0.025325234	0.257581515

CRITICAL COMMENT

There are some essential steps well executed in this process of building models. Basically, the main intended purpose of this practical is achieved, models predicting output variables are simply trained and optimized. However, the performance of the models is not quite virtue. There are some possible reasons:

1. Although there are two optimization strategies applied, optimization may still be not suitable and efficient enough.
2. It is possible that the assumed multi-feature models are not able to predict the output variables efficiently. Polynomial regression model may have better performance in this case.
3. Sample size is insufficient in extent.

Besides, data preparation does not involve any advanced statistical methods, which may make training not quite efficient.

To summarize, this is a well-executed model building process but there are still a lot of room of improvement.

EXTENSION: LOCALLY WEIGHTED LINEAR REGRESSION

Introduction

As an effective method to solve the underfitting problem of linear regression, locally weighted linear regression (LWLR) algorithm predicts the value of the output variables by a non-parametric model.

To build and optimize a linear regression model, it is necessary to calculate sum of squared errors minimize it. However, in LWLR algorithm, a weight is added to each square error, and what the algorithm do is to minimize the weighted squared errors. [3]

Hence the loss function is changed to:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m} w^{(i)}(\theta^T x^{(i)} - y^{(i)})^2$$

$w^{(i)}$ is the weight of i th observation and its value is calculated by:

$$w^{(i)} = \exp\left(-\frac{\left(x^{(i)} - x\right)^2}{2\tau^2}\right)$$

From this equation, it could be known that observation whose input value is closer to the test input would have larger weight.

Since the normal equation is derived from the loss function, so the normal equation is now changed to:

$$\theta = (X^T W X)^{-1} X^T W y$$

W is a diagonal matrix whose (i, i) is $w^{(i)}$. That means it need to use a set of data to predict output for a specific group of input variables.

Back to this practical, since there are no parametrical models assumed, the algorithm predicts outputs for the validation set with observations in the training set. The value of τ is initially set to 1 and would change to optimize the models. However, in this practical, the value of τ is fixed to 0.1 because it is time-consuming to evaluate and optimize it with hundreds of test samples. Instead, to compare LWLR with the parametric models trained by normal equation before, 20 groups of input variables (10 to predict Y1 and Y2 respectively) are randomly selected from validation set, LWLR and normal equation models predict the output values respectively. Results are listed in Table 5.

Table 5. (a) Prediction for Y1

Observation value	Prediction of LWLR	Prediction of parametric model
-0.253362106	-0.259780905	-0.138369115
-0.310250755	-0.275226296	-0.129474916
0.256209206	0.170734273	0.145770814
-0.204831505	-0.275226296	-0.129474916
-0.252283648	-0.259780905	-0.138369115
0.052650296	0.147171286	0.172331805
0.127333499	0.170734273	0.145770814
-0.329662995	-0.282516366	-0.120600984
-0.212111095	-0.282516366	-0.120600984
-0.192698854	-0.198868586	-0.156137247

(b) Prediction for Y2

Observation value	Prediction of LWLR	Prediction of parametric model
-0.248256407	-0.234574961	-0.16976170
-0.277612723	-0.272022321	-0.108041939
0.178083479	0.184646977	0.169651650
0.341563145	0.251359372	0.319990976
0.319209253	0.326963571	0.072566313
0.042882832	0.084612140	0.488013089
0.312745477	0.184646977	0.169651650
-0.306699714	-0.307854465	-0.046231241
0.143879331	0.091761498	0.107750017
-0.146721260	-0.162222376	-0.293292161

Table 5 shows that most predictions made by LWLR are more accurate than those made by the parametric models trained by normal equation. High accuracy is the advantage of LWLR.

However, a vital problem of efficiency appears with high accuracy. Since accuracy is improved because the algorithm always uses the whole train set to predict every output, which means the time complexity of the algorithm is extremely high. Moreover, it may cause a shortage of local memory if it is asked to predict for a large set of inputs, which actually happened in this practical.

Besides, another drawback of LWLR may be that it could not work if the test input is not in the range covered by the training data. Furthermore, outliers may also pose a threat to the accuracy of results.

Based on the experiment and the above discussion, it could be safe to draw a conclusion that LWLR would performance better than some parametric linear models if the test set is not very large. However, low universality and low efficiency would make it not suitable for large dataset.

PREFERENCE

1. Wickham H. Tidy data[J]. Journal of Statistical Software, 2014, 59(10): 1-23.
2. Tsanas A, Xifara A. Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools[J]. Energy and Buildings, 2012, 49: 560-567.
3. Tomáš Bouda. Day 97: Locally weighted regression. 100 Days of Algorithm. Available from:

   https://medium.com/100-days-of-algorithms/day-97-locally-weighted-regression-c9cfaff087fb