Statistical learning: First assignment

Ali Zamani(96123035)

November 3, 2018

0.1 Section 1

0.1.1 1)

The advantages of a very flexible approach are that it may give a better fit for non-linear models and it decreases the bias.
The disadvantages of a very flexible approach are that it requires estimating a greater number of parameters, it follows the noise too closely and it increases the variance.
A more flexible approach would be preferd to a less flexible approach when wa are interested in prediction and not the interpretability of the results.
A less flexible approach would be preferd a more flexible approach when we are interested in inference and the interpretability of the results.

0.1.2 2)

A confidence interval gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data.
If the 95% confidence interval contains zero (more precisely, the parameter value specified in the null hypothesis), then the effect will not be significant at the 0.05 level. Looking at non-significant effects in terms of confidence intervals makes clear why the null hypothesis should not be accepted when it is not rejected: Every value in the confidence interval is a plausible value of the parameter. Since zero is in the interval, it cannot be rejected. However, there is an infinite number of other values in the interval (assuming continuous measurement), and none of them can be rejected either.

0.1.3 3)

a)

$$\begin{aligned}
E_{x,y}\left[\left(y - w^T x\right)^2\right] &= \\
E_{x,y}\left[y^2 + \left(w^T x\right)\left(x^T w\right) - 2yw^T x\right] &= \\
E_y\left[y^2\right] + w^T E_x\left[xx^T\right] w - 2E_{x,y}\left[w^T yx\right] &= \\
E_y\left[y^2\right] + w^T Rw - 2w^T c &\rightarrow \\
\frac{\partial(E_y\left[y^2\right] + w^T Rw - 2w^T c)}{\partial w} &= 0 \rightarrow \\
2Rw - 2c = 0 \rightarrow w &= R^{-1} c
\end{aligned} \tag{1}$$

b)

$$\begin{aligned}
E_{x,y}\left[\left(y-\tilde{w}^T x\right)^2\right] &= \\
E_{x,y}\left[\left(y-\tilde{w}^T x + w^{\star T}x - w^{\star T}x\right)^2\right] &= \\
E_{x,y}\left[\left(y-w^{\star T}x\right)^2 + \left(w^{\star T}x-\tilde{w}^T x\right)^2 + 2\left(y-w^{\star T}x\right)\left(w^{\star T}x-\tilde{w}^T x\right)\right] &= \\
E_{x,y}\left[\left(y-w^{\star T}x\right)^2\right] + E_x\left[\left(w^{\star T}x-\tilde{w}^T x\right)^2\right]
\end{aligned}$$

The error of estimation is the summation of irreducible error(first term) and reducible error(second term).

c)

The first term is irreducible error and the second term is the reducible error.

0.1.4 4)

I.

M=0: this model only considers the average of given n sample so it is underfitted.
M=1: the value of w is near together hence it seems that it is the best model.
M=4 : the values of w near together and don't change frequently hence it seems that it is better than model whit M=1
M=9: the values of w don't near together and frequently change between positive and negative hence they are overfitted.

II.

when M is changing from 0 to 4, the model becomes fit hence the training and testing error become decrease.
when M is changing from 4 to 9, the model becomes overfit so the training error becomes decrease and the test error becomes increase.

III.

In order to improve the regression model, we can increase the number of sampling data if it is possible or use Multiplication of features as an input of the linear regression model.

IV.

Increasing N avoids overfitting as order of polynomial increase so the testing error decreases. We cannot make any claim about training error.

0.1.5 5)

$$\hat{y}_i = x_i \frac{\sum_{j=1}^n x_j y_j}{\sum_{k=1}^n x_k^2} = \sum_{j=1}^n \frac{x_i x_j}{\sum_{k=1}^n x_k^2} y_j = \sum_{j=1}^n a_j y_j \rightarrow$$
$$a_j = \frac{x_i x_j}{\sum_{k=1}^n x_k^2}$$

0.1.6 6)

We know that:

$$\beta^\star = (X^T X)^{-1} X^T Y \rightarrow \beta^\star = \frac{\sum_{j=1}^n x_j y_j}{\sum_{k=1}^n x_k^2}$$

$$\hat{y}_i = \hat{\beta}_1 x_i$$

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$R^2 = 1 - \frac{\sum_i (y_i - (\sum_j x_j y_j / \sum_j x_j^2) x_i)^2}{\sum_j y_j^2} =$$
$$\frac{\sum_j y_j^2 - (\sum_i y_i^2 - 2\sum_i y_i (\sum_j x_j y_j / \sum_j x_j^2) x_i + \sum_i (\sum_j x_j y_j / \sum_j x_j^2)^2 x_i^2)}{\sum_j y_j^2} \rightarrow$$
$$R^2 = \frac{2(\sum_i x_i y_i)^2 / \sum_j x_j^2 - (\sum_i x_i y_i)^2 / \sum_j x_j^2}{\sum_j y_j^2} = \frac{(\sum_i x_i y_i)^2}{\sum_j x_j^2 \sum_j y_j^2} = Cor(X,Y)^2.$$

0.2 Section2

Code:

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Fri Nov  2 16:38:09 2018

@author: Ali Zamani---zamaniali1995@gmail.com
"""
# Load libraries
import pandas
import matplotlib.pyplot as plt
from tabulate import tabulate
from pandas.plotting import scatter_matrix
import statsmodels.formula.api as smf
```

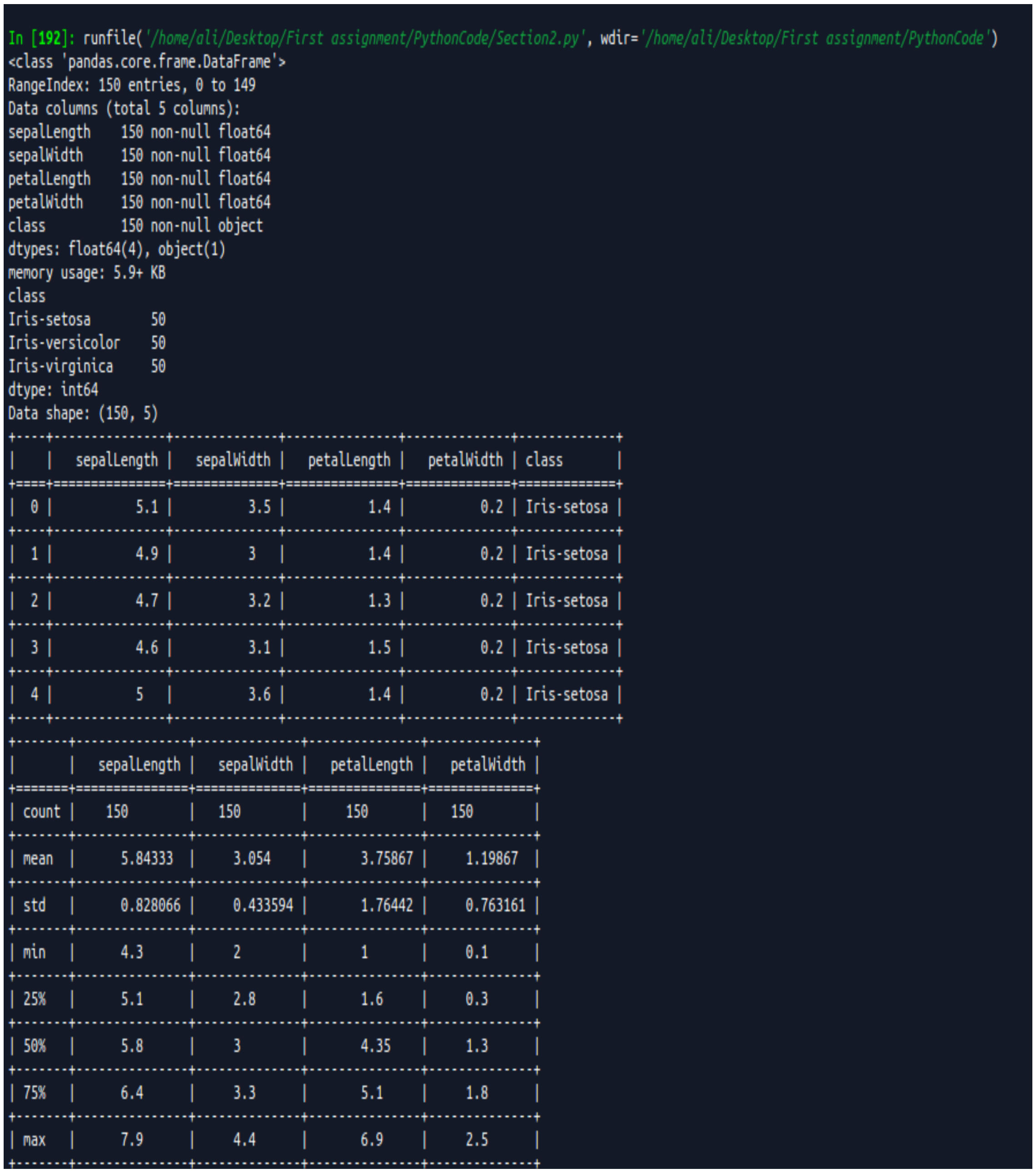
```
from sklearn.linear_model import LinearRegression
plt.style.use('seaborn-white')
#%%
# Load dataset
names = ['sepalLength', 'sepalWidth', 'petalLength', 'petalWidth',
    'class']
names1 = ['','sepalLength', 'sepalWidth', 'petalLength', '
    petalWidth']
dataSet = pandas.read_csv('Data/iris.csv', names=names,usecols
    =[0,1,2,3,4])
dataSet.info()
print(dataSet.groupby('class').size())
print ("Data shape:", dataSet.shape)
print (tabulate(dataSet.head(5),headers=names,tablefmt="grid"))
print (tabulate(dataSet.describe(),headers=names1,tablefmt="grid"))
#print(dataset.describe())
#%%
#scatter plot matrix
scatter_matrix(dataSet,grid=True)
#%%
#Linear Regression(petalLength as a fuction of petalWidth)
petalWidth=dataSet.values[:,3].reshape(-1,1)
petalLength=dataSet.values[:,2].reshape(-1,1)
LR= LinearRegression()
LR.fit(petalWidth,petalLength)
print ("Intercept:" , LR.intercept_)
print ("Slope:" , LR.coef_)
petalLength_pred=LR.predict(petalWidth)
LRfig, LRax = plt.subplots( nrows=1, ncols=1 )
LRax.scatter(petalWidth,petalLength)
LRax.plot(petalWidth,petalLength_pred,color='red')
LRfig.savefig('LR.pdf')
#Report the t-value and p-value
est=smf.ols('petalLength ~ petalWidth', dataSet).fit()
print (est.summary().tables[1])

#%%
sepalWidth=dataSet.values[:,1].reshape(-1,1)
V=petalWidth*sepalWidth
LR_mul= LinearRegression()
LR_mul.fit(V,petalLength)
print ("Intercept:" , LR_mul.intercept_)
print ("Slope:" , LR_mul.coef_)
petalLenMul_pred=LR_mul.predict(V)
LR_mulfig, LR_mulax = plt.subplots( nrows=1, ncols=1 )
LR_mulax.scatter(V,petalLength)
LR_mulax.plot(V,petalLenMul_pred,color='red')
LR_mulfig.savefig('LR_mul.pdf')
#Report the t-value and p-value
est_mul=smf.ols('petalLength ~ petalWidth+sepalWidth+petalWidth*
    sepalWidth', dataSet).fit()
print (est_mul.summary().tables[1])

dataSet.corr()
```

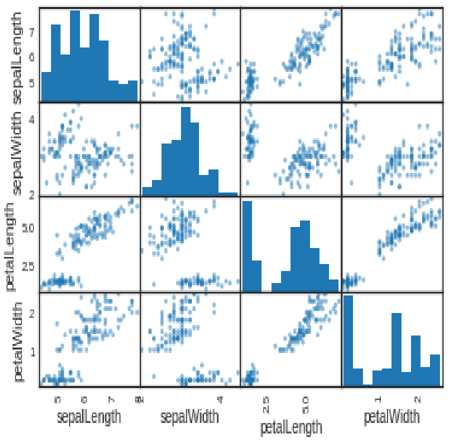
Results:

A,B)

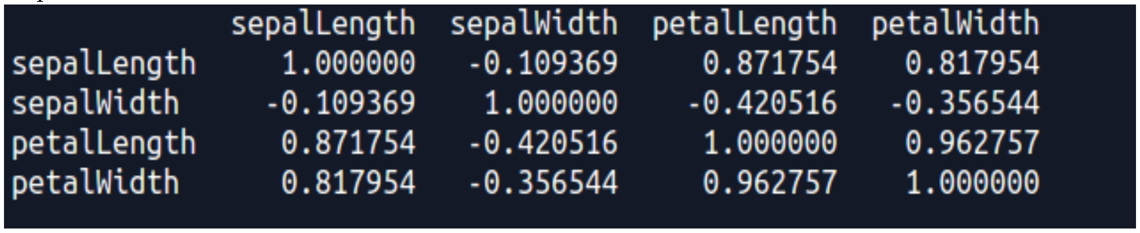


```
In [192]: runfile('/home/ali/Desktop/First assignment/PythonCode/Section2.py', wdir='/home/ali/Desktop/First assignment/PythonCode')
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepalLength    150 non-null float64
sepalWidth     150 non-null float64
petalLength    150 non-null float64
petalWidth     150 non-null float64
class          150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
Data shape: (150, 5)
+----+---------------+--------------+---------------+--------------+-------------+
|    |   sepalLength |   sepalWidth |   petalLength |   petalWidth | class       |
+====+===============+==============+===============+==============+=============+
|  0 |           5.1 |          3.5 |           1.4 |          0.2 | Iris-setosa |
+----+---------------+--------------+---------------+--------------+-------------+
|  1 |           4.9 |          3   |           1.4 |          0.2 | Iris-setosa |
+----+---------------+--------------+---------------+--------------+-------------+
|  2 |           4.7 |          3.2 |           1.3 |          0.2 | Iris-setosa |
+----+---------------+--------------+---------------+--------------+-------------+
|  3 |           4.6 |          3.1 |           1.5 |          0.2 | Iris-setosa |
+----+---------------+--------------+---------------+--------------+-------------+
|  4 |           5   |          3.6 |           1.4 |          0.2 | Iris-setosa |
+----+---------------+--------------+---------------+--------------+-------------+
+-------+---------------+--------------+---------------+--------------+
|       |   sepalLength |   sepalWidth |   petalLength |   petalWidth |
+=======+===============+==============+===============+==============+
| count |    150        |   150        |    150        |   150        |
+-------+---------------+--------------+---------------+--------------+
| mean  |      5.84333  |     3.054    |      3.75867  |     1.19867  |
+-------+---------------+--------------+---------------+--------------+
| std   |      0.828066 |     0.433594 |      1.76442  |     0.763161 |
+-------+---------------+--------------+---------------+--------------+
| min   |      4.3      |     2        |      1        |     0.1      |
+-------+---------------+--------------+---------------+--------------+
| 25%   |      5.1      |     2.8      |      1.6      |     0.3      |
+-------+---------------+--------------+---------------+--------------+
| 50%   |      5.8      |     3        |      4.35     |     1.3      |
+-------+---------------+--------------+---------------+--------------+
| 75%   |      6.4      |     3.3      |      5.1      |     1.8      |
+-------+---------------+--------------+---------------+--------------+
| max   |      7.9      |     4.4      |      6.9      |     2.5      |
+-------+---------------+--------------+---------------+--------------+
```

C)

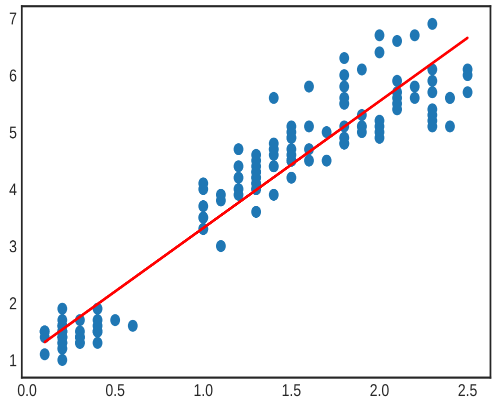


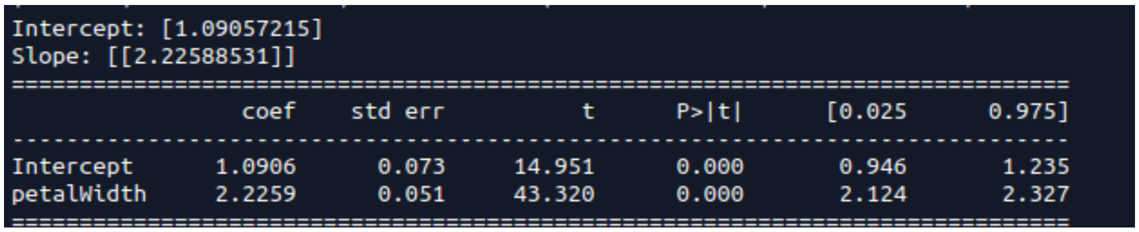
Explanation of the relation of features:



```
             sepalLength  sepalWidth  petalLength  petalWidth
sepalLength     1.000000   -0.109369     0.871754    0.817954
sepalWidth     -0.109369    1.000000    -0.420516   -0.356544
petalLength     0.871754   -0.420516     1.000000    0.962757
petalWidth      0.817954   -0.356544     0.962757    1.000000
```

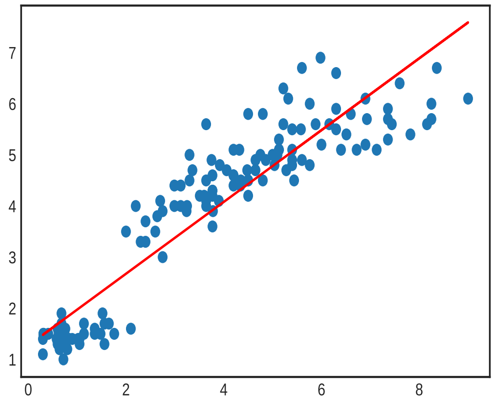
D)

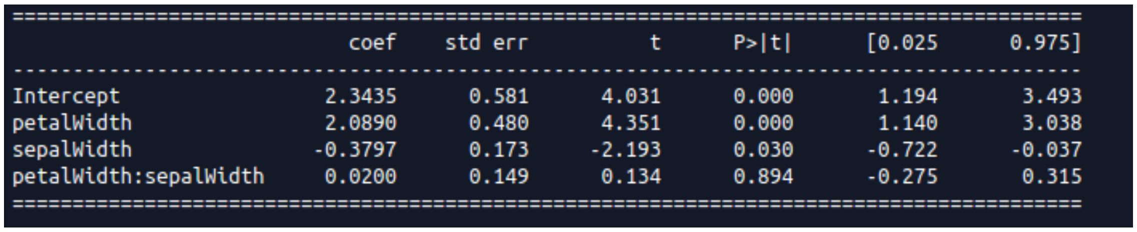




```
Intercept: [1.09057215]
Slope: [[2.22588531]]
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      1.0906      0.073     14.951      0.000       0.946       1.235
petalWidth     2.2259      0.051     43.320      0.000       2.124       2.327
==============================================================================
```

E)





```
==================================================================================
                           coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept                2.3435      0.581      4.031      0.000       1.194       3.493
petalWidth               2.0890      0.480      4.351      0.000       1.140       3.038
sepalWidth              -0.3797      0.173     -2.193      0.030      -0.722      -0.037
petalWidth:sepalWidth    0.0200      0.149      0.134      0.894      -0.275       0.315
==================================================================================
```

The p-value of petalWidth*sepalWidth is 0.894 which is large so we conclude that feature petalWidth*sepalWidth cannot improve our model.
The confidence interval of petalWidth*sepalWidth is from -0.275 to 0.315 which is consists of 0 so confidence interval verify that petalWidth*sepalWidth does not improve our model either.

Source Code