HW 2

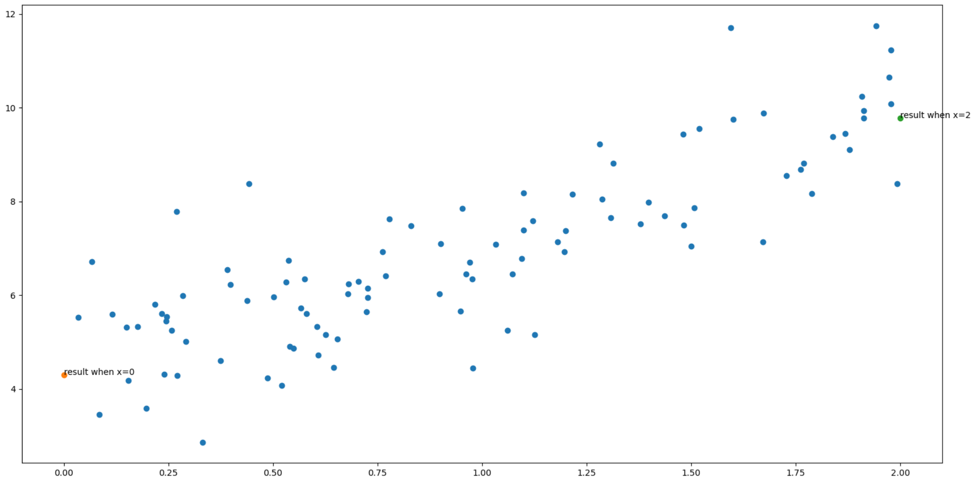
2.1

> 实验2-1:编写程序，通过上一页产生的附加噪声的随机数据,
>
> (1）做线性回归(即求出θ)
>
> (2)利用取得的回归模型,对x=0和x=2两个点做预测
>
> (3)对上述随机数据和预测结果利用plot可视化。

使用如下代码进行拟合

```
import numpy as np
from sympy import *
import matplotlib.pyplot as plt

import io
import sys

from numpy.lib.function_base import average
sys.stdout = io.TextIOWrapper(sys.stdout.buffer,encoding='utf-8')

x = 2*np.random.rand(100,1)
y = 4+3*x+np.random.randn(100,1)

tmp = np.ones(100)
xTmp = np.insert(x,0,tmp,axis=1)

#对x，y进行线性回归
X = np.matrix(xTmp)
res = ((X.I))*y
print("结果矩阵为:",res)
print("常数项为:",float(res[0]))
print("系数为:",float(res[1]))
print("x=0时的预测结果为:",float(res[0])+0*float(res[1]))
print("x=2时的预测结果为:",float(res[0])+2*float(res[1]))

#画图
plt.scatter(x,y)
plt.scatter(0,float(res[0])+0*float(res[1]))
plt.text(0,float(res[0])+0*float(res[1]),"result when x=0")
plt.scatter(2,float(res[0])+2*float(res[1]))
plt.text(2,float(res[0])+2*float(res[1]),"result when x=2")
plt.show()
```

得到如下结果



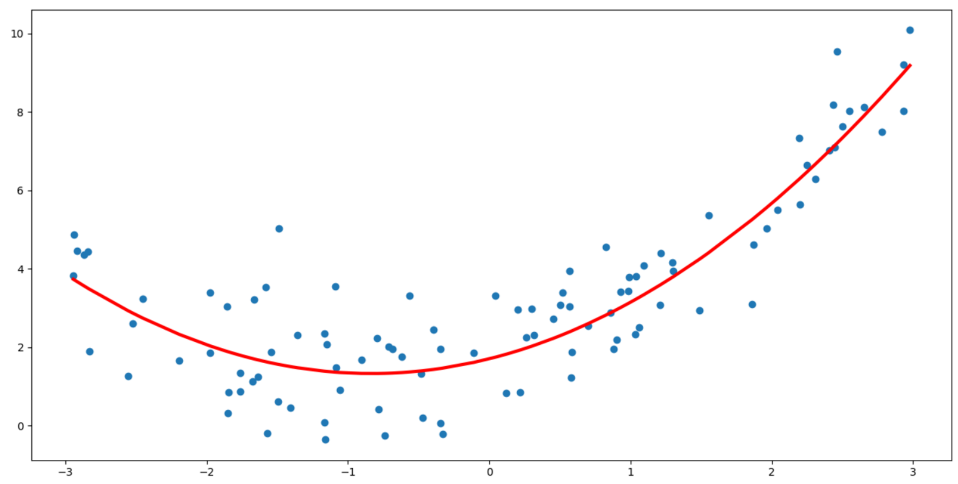
2.2

> 实验2-2:通过网上调研了解Scikit-Learn中的LinearRegression模块，编写程序，利用LinearRegression模块:
>
> (1)对上述产生的二级多项式数据(x_poly,y)做线性回归。
> (2)可视化回归模型和数据。

使用如下代码进行拟合

```
from sklearn.preprocessing import PolynomialFeatures
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt

m=100
X = 6*np.random.rand(m,1) - 3
Y = 0.5*X**2 + X + 2 + np.random.randn(m,1)
lr=linear_model.LinearRegression()

poly_feacures = PolynomialFeatures(degree=2,include_bias=False)
X_poly = poly_feacures.fit_transform(X)

lr.fit(X_poly,Y)

X = np.sort(X,axis=0)
Y = 0.5*X**2 + X + 2 + np.random.randn(m,1)
print(X)

plt.scatter(X,Y)
plt.plot(X,lr.predict(poly_feacures.fit_transform(X)),color='red',linewidth=3)
plt.show()
```

得到的结果如下:



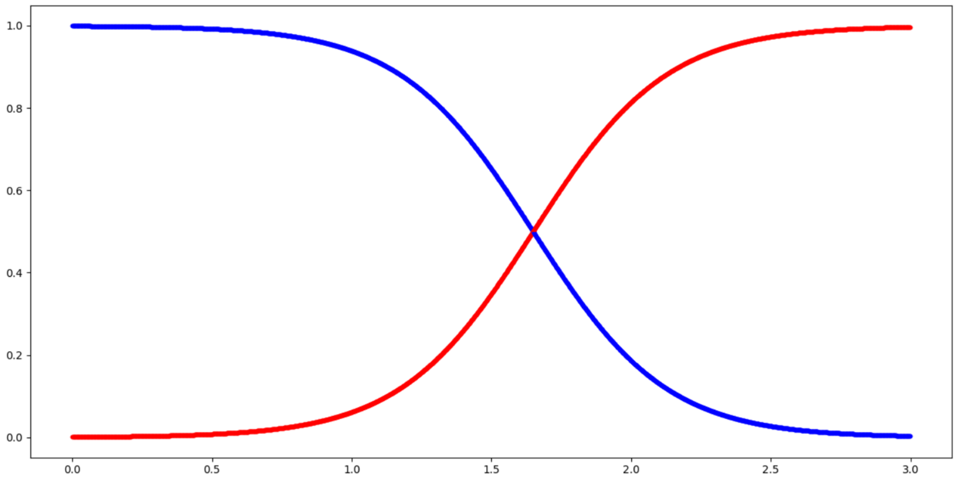
2.3

> 实验2-3:查阅网上资料，通过sklearn.linear_model中的LogisticRegression，利用花瓣宽度,实现一个分类器检测维吉亚鸢尾花。
> 可视化花瓣宽度0-3 cm之间的鸢尾花模型估算出的概率。

使用如下代码

```
from statistics import mode
from turtle import color
from sklearn import datasets
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


iris = datasets.load_iris()
X = iris["data"][:,3:]
Y = (iris["target"] == 2).astype(np.int)#1 if target , 0 else

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1,
random_state=0)

model = linear_model.LogisticRegression()
model.fit(X_train, Y_train)

truthRate = model.score(X_test, Y_test)

print(truthRate)

for i in np.arange(0,3,0.001):
    plt.plot(i,model.predict_proba(i.reshape(1,-1))[0][0],'b.')
    plt.plot(i,model.predict_proba(i.reshape(1,-1))[0][1],'r.')
plt.show()
```

得到如下结果


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0