法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

回归实践

邹博

主要内容

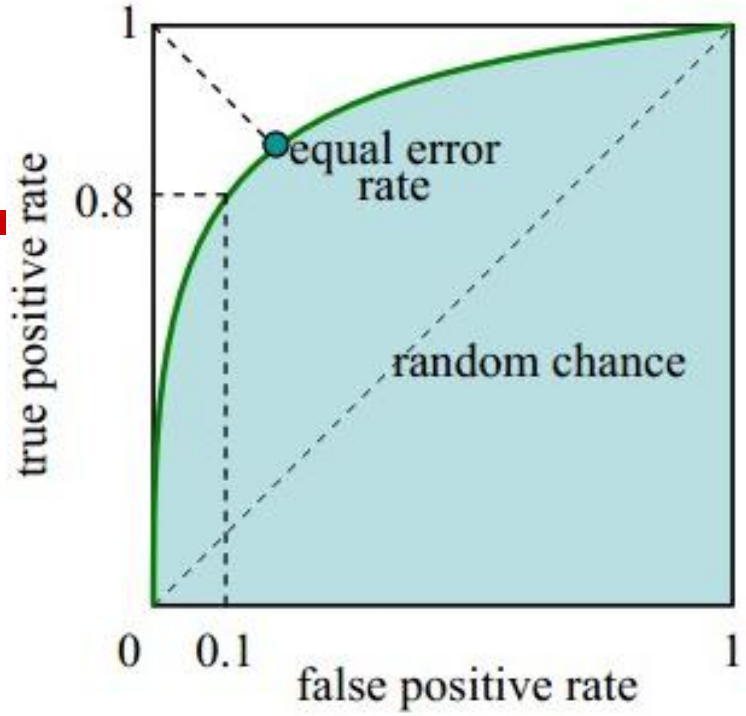
- AUC
  - 分类器指标
- 代码实践
  - 调参与交叉验证
- 该部分PPT中仅列举模型效果截图，详细内容请参考该PPT的配套代码。

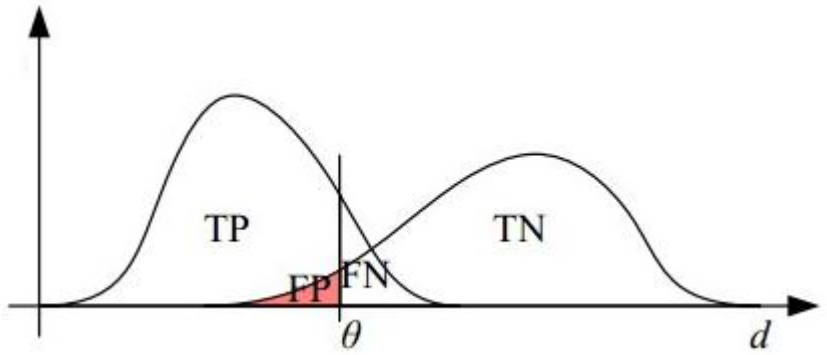
AUC

预测值 / 实际值	Positive	Negtive
正	TP	FN
负	FP	TN

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$



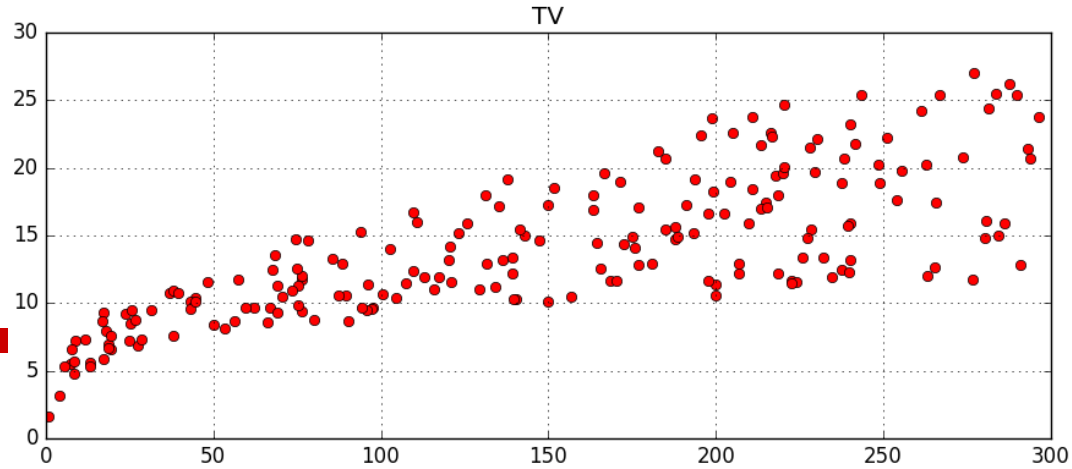


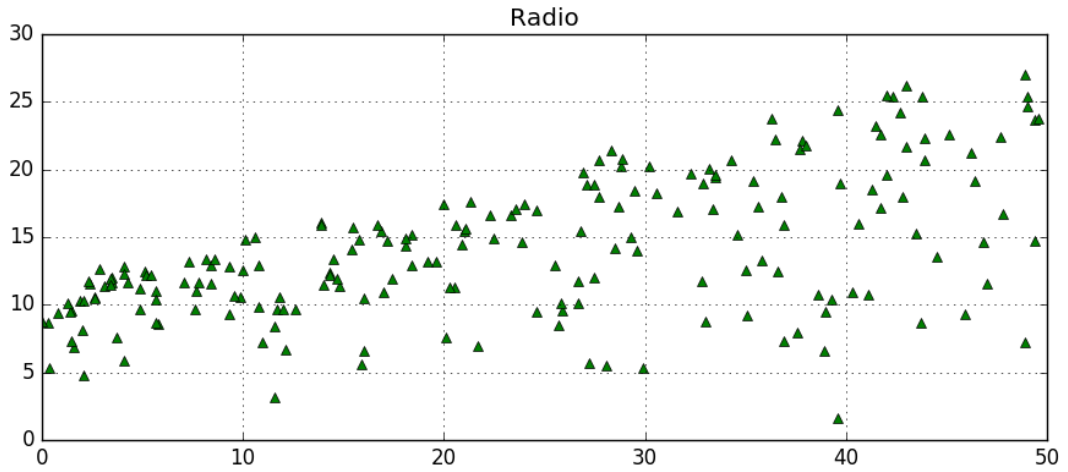
Receiver Operating Characteristic

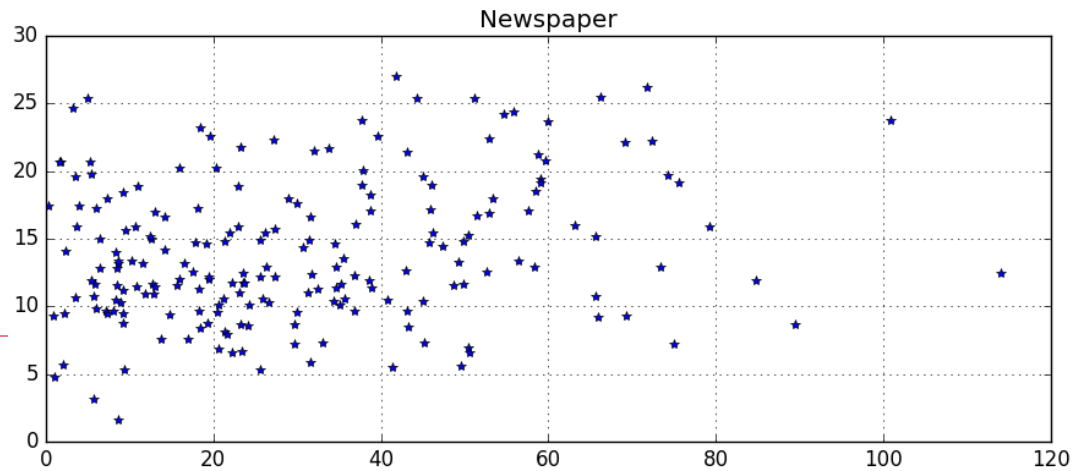
Area Under Curve

数据显示

	TV	Radio	Newspaper	Sales
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
6	8.7	48.9	75	7.2
7	57.5	32.8	23.5	11.8
8	120.2	19.6	11.6	13.2
9	8.6	2.1	1	4.8
10	199.8	2.6	21.2	10.6
11	66.1	5.8	24.2	8.6
12	214.7	24	4	17.4
13	23.8	35.1	65.9	9.2
14	97.5	7.6	7.2	9.7
15	204.1	32.9	46	19
16	195.4	47.7	52.9	22.4
17	67.8	36.6	114	12.5
18	281.4	39.6	55.8	24.4
19	69.2	20.5	18.3	11.3
20	147.3	23.9	19.1	14.6
21	218.4	27.7	53.4	18
22	237.4	5.1	23.5	12.5
23	13.2	15.9	49.6	5.6
24	228.3	16.9	26.2	15.5
25	62.3	12.6	18.3	9.7
26	262.9	3.5	19.5	12
27	142.9	29.3	12.6	15
28	240.1	16.7	22.9	15.9
29	248.8	27.1	22.9	18.9
30	70.6	16	40.8	10.5
31	292.9	28.3	43.2	21.4
32	112.9	17.4	38.6	11.9
33	97.2	1.5	30	9.6
34	265.6	20	0.3	17.4
35	95.7	1.4	7.4	9.5
36	290.7	4.1	8.5	12.8
37	266.9	43.8	5	25.4
38	74.7	49.4	45.7	14.7
39	43.1	26.7	35.1	10.1
40	228	37.7	32	21.5

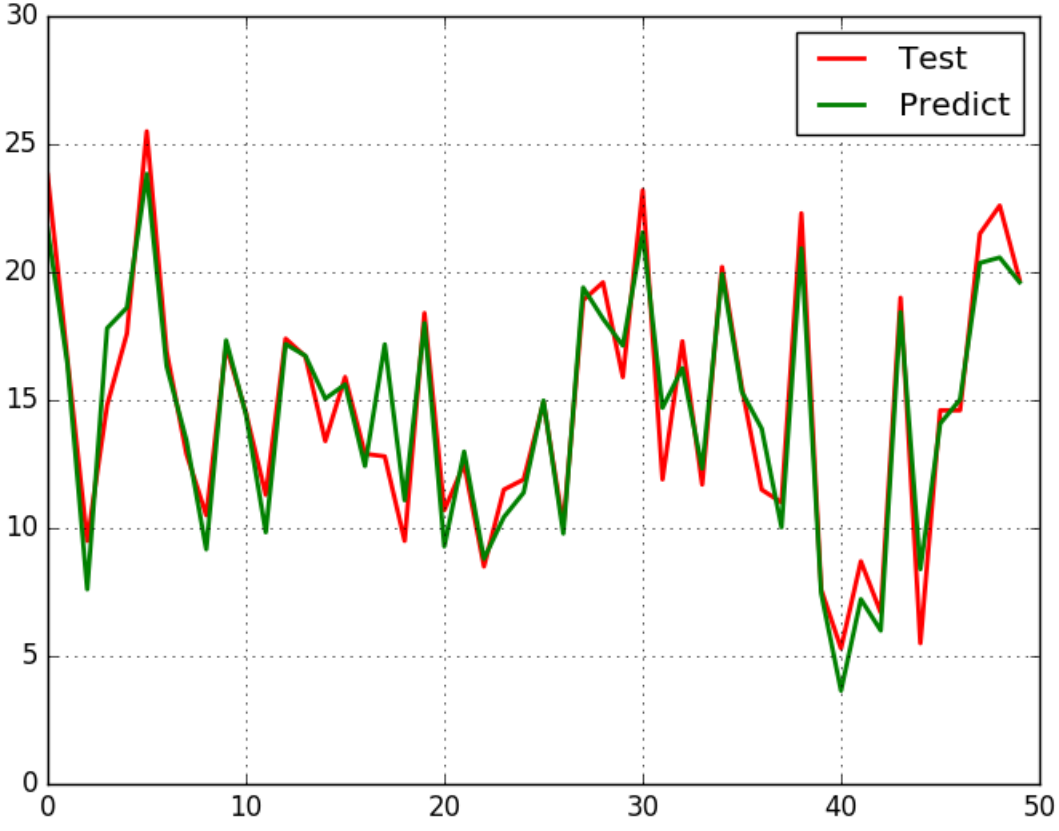






拟合与预测

- y=2.877+0.046*TV+0.179*Radio+0.0035*Newspaper



波士顿房屋价格预测

- 波士顿房价数据最早来自于卡耐基梅隆大学CMU的统计图书馆(StatLib library)，由Harrison D.和Rubinfeld D.L在1978年的著作Hedonic prices and the demand for clean air中。
  - 数据下载链接：https://archive.ics.uci.edu/ml/datasets/Housing
- 特征描述：
  1. CRIM: per capita crime rate by town
  2. ZN: proportion of residential land zoned for lots over 25,000 sq.ft.
  3. INDUS: proportion of non-retail business acres per town
  4. CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
  5. NOX: nitric oxides concentration (parts per 10 million)
  6. RM: average number of rooms per dwelling
  7. AGE: proportion of owner-occupied units built prior to 1940
  8. DIS: weighted distances to five Boston employment centres
  9. RAD: index of accessibility to radial highways
  10. TAX: full-value property-tax rate per $10,000
  11. PTRATIO: pupil-teacher ratio by town
  12. B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
  13. LSTAT: % lower status of the population
  14. MEDV: Median value of owner-occupied homes in $1000's

Elastic Net/LASSO的2阶特征预测

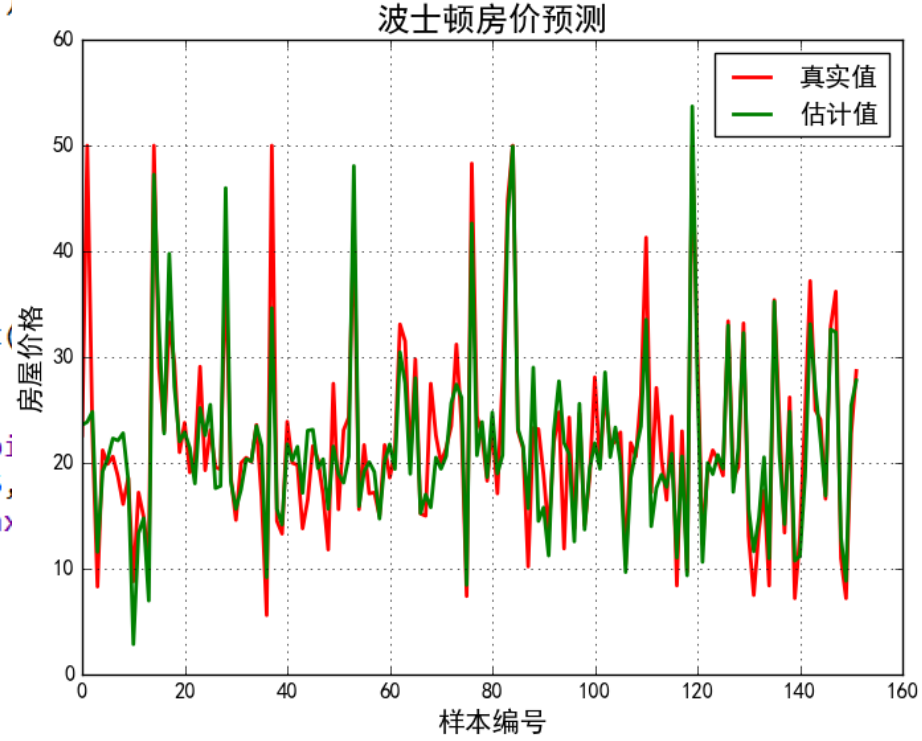
```
file_data = pd.read_csv('8.housing.data', header=None
# a = np.array([float(s) for s in str if s != ''])
data = np.empty((len(file_data), 14))
for i, d in enumerate(file_data.values):
    d = map(float, filter(not_empty, d[0].split(' ')
    data[i] = d
x, y = np.split(data, (13, ), axis=1)
# data = sklearn.datasets.load_boston()
# x = np.array(data.data)
# y = np.array(data.target)
print u'样本个数：%d，特征个数：%d' % x.shape
print y.shape

x_train, x_test, y_train, y_test = train_test_split(
model = Pipeline([
    ('ss', StandardScaler()),
    ('poly', PolynomialFeatures(degree=3, include_bi
    ('linear', ElasticNetCV(l1_ratio=[0.1, 0.3, 0.5,
                            fit_intercept=False, max
])
model.fit(x_train, y_train.ravel())
linear = model.get_params('linear')['linear']
print u'超参数：', linear.alpha_
print u'L1 ratio：', linear.l1_ratio_
```

超参数： 0.16681005372

L1 ratio： 1.0

R2: 0.79803070365

均方误差： 16.8170747774



小结

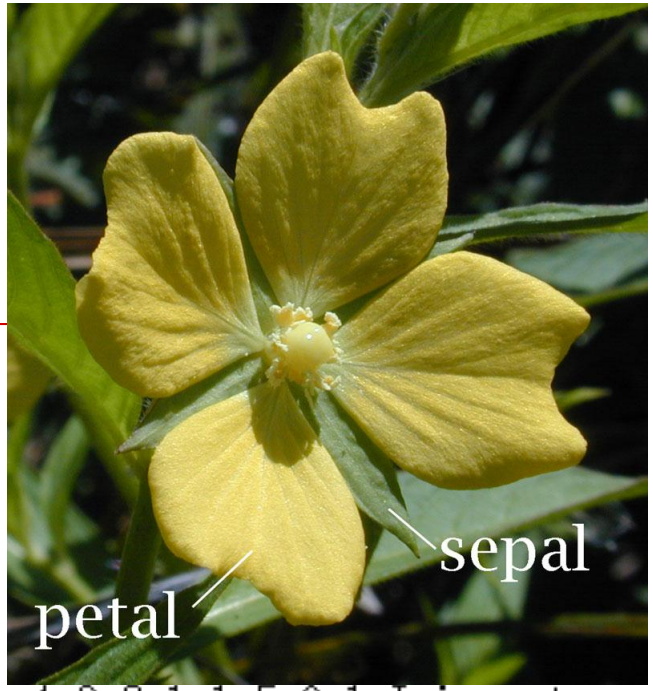
- 本模型虽然简单，但它涵盖了机器学习的相当部分的内容。
  - 使用75%的训练集和25%的测试集
  - 分析模型后，使用最为简单的方法：直接删除；得到了更好的预测结果。
- 奥卡姆剃刀
  - 如果能够用简单模型解决问题，则不使用更为复杂的模型。因为复杂模型往往增加不确定性，造成过多人力和物力成本，且容易过拟合。

鸢尾花数据集

- 鸢尾花数据集或许是最有名的模式识别测试数据。
  - 早在1936年，模式识别的先驱Fisher就在论文“The use of multiple measurements in taxonomic problems”中使用了它(直至今日该论文仍然被频繁引用)。
- 该数据集包括3个鸢尾花类别，每个类别有50个样本。其中一个类别是与另外两类线性可分的，而另外两类不能线性可分。
  - 由于Fisher的最原始数据集存在两个错误(35号和38号样本)，实验中我们使用的是修正过的数据。
- 下载链接：http://archive.ics.uci.edu/ml/datasets/Iris

数据描述

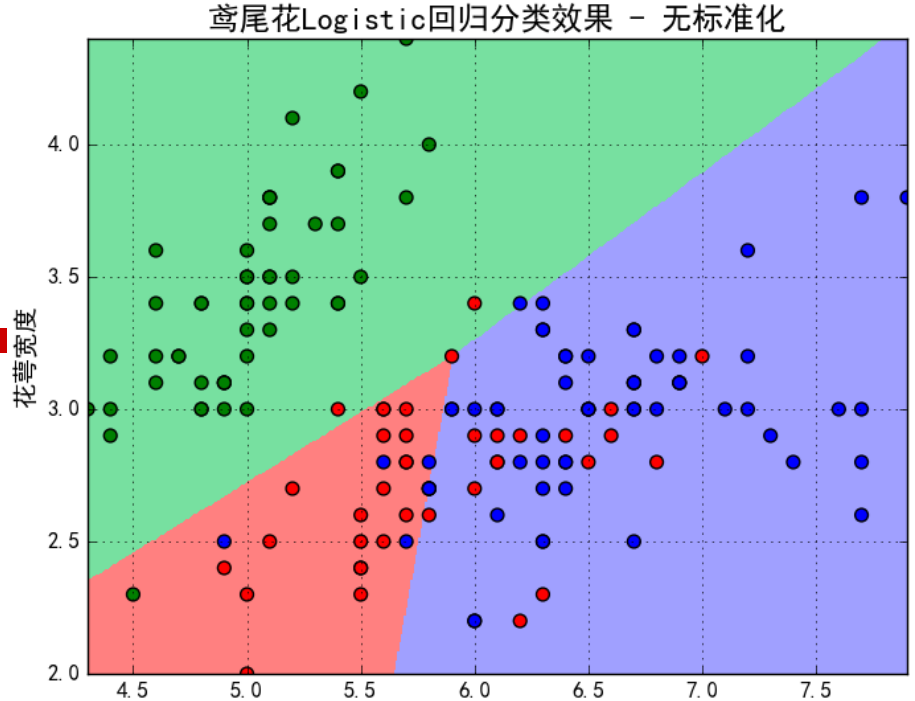
- 该数据集共150行，每行1个样本。每个样本有5个字段，分别是
  - 花萼长度(单位cm)
  - 花萼宽度(单位：cm)
  - 花瓣长度(单位：cm)
  - 花瓣宽度(单位：cm)
  - 类别(共3类)
    - Iris Setosa
    - Iris Versicolour
    - Iris Virginica

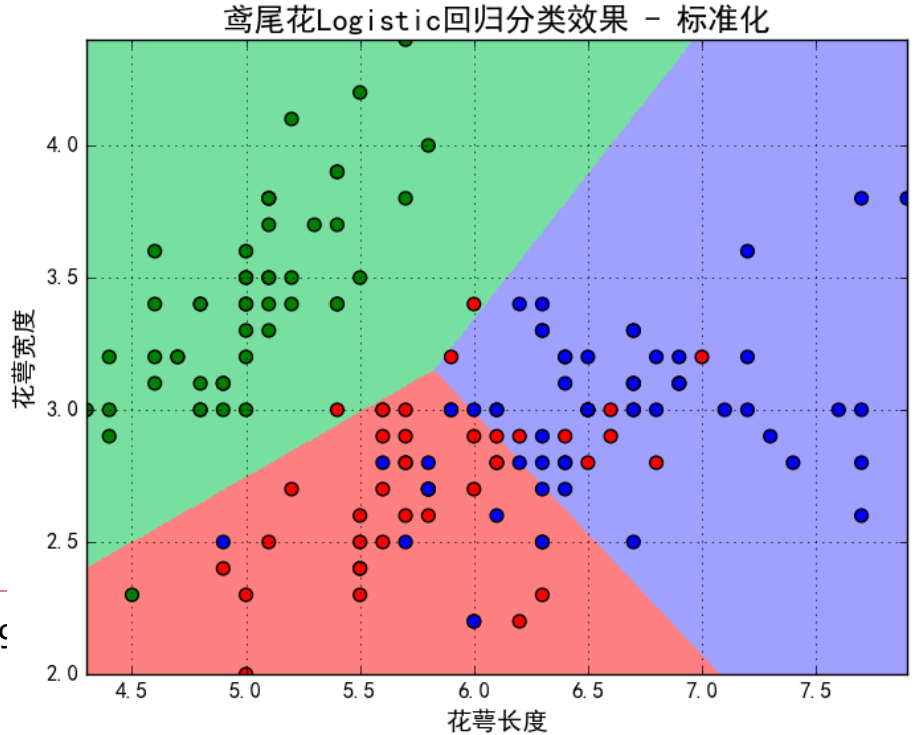



```
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
5.1,3.8,1.5,0.3,Iris-setosa
5.4,3.4,1.7,0.2,Iris-setosa
5.1,3.7,1.5,0.4,Iris-setosa
4.6,3.6,1.0,0.2,Iris-setosa
```

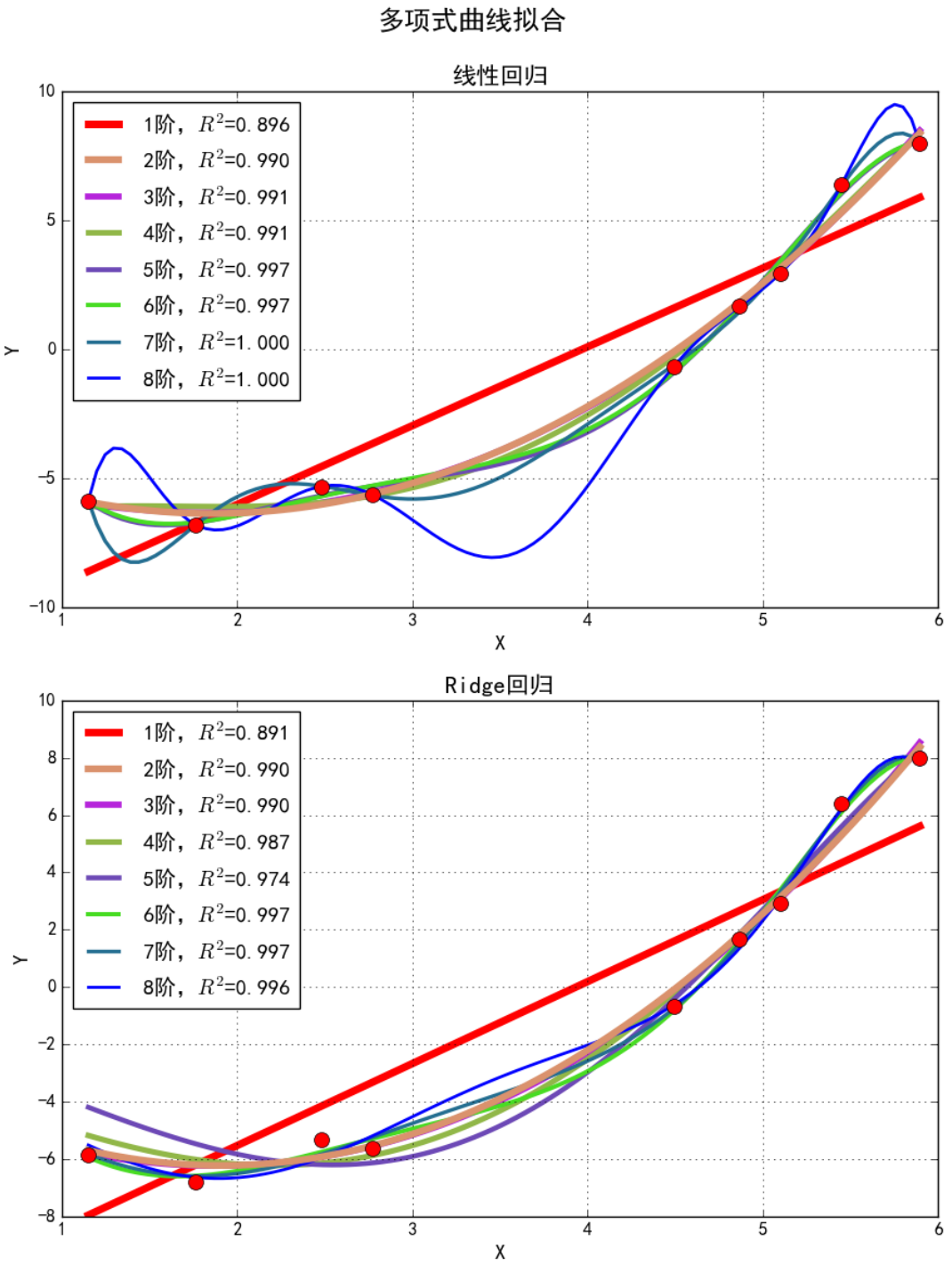
鸢尾花的分类

```
def iris_type(s):
    it = {'Iris-setosa': 0,
          'Iris-versicolor': 1,
          'Iris-virginica': 2}
    return it[s]

# 路径，浮点型数据，逗号分隔，第4列使用函数iris_type单独处理
data = np.loadtxt(path, dtype=float, delimiter=',',
                  converters={4: iris_type})
x, y = np.split(data, (4,), axis=1)
print 'x = \n', x
print 'y = \n', y
# 仅使用前两列特征
x = x[:, :2]
lr = Pipeline([('sc', StandardScaler()),
               ('clf', LogisticRegression()) ])
lr.fit(x, y.ravel())
y_hat = lr.predict(x)
print u'准确度：%.2f%%' % (100*np.mean(y_hat == y.ravel()))
# 画图
N, M = 500, 500
```

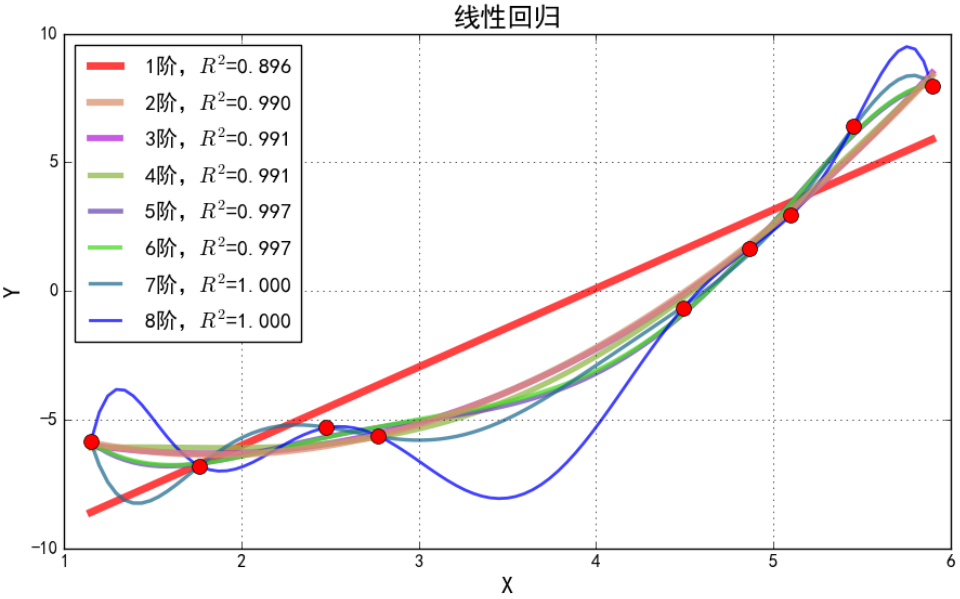


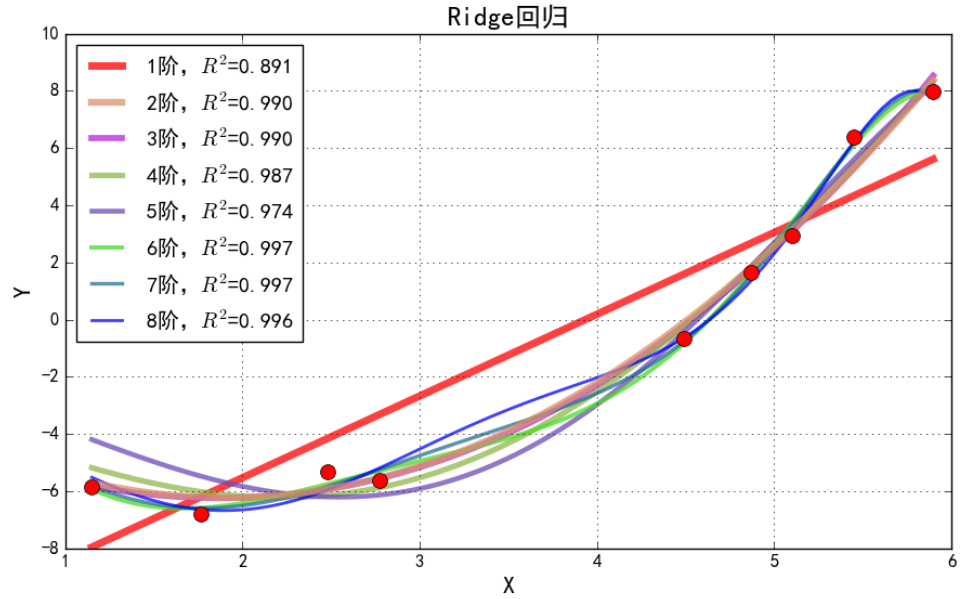


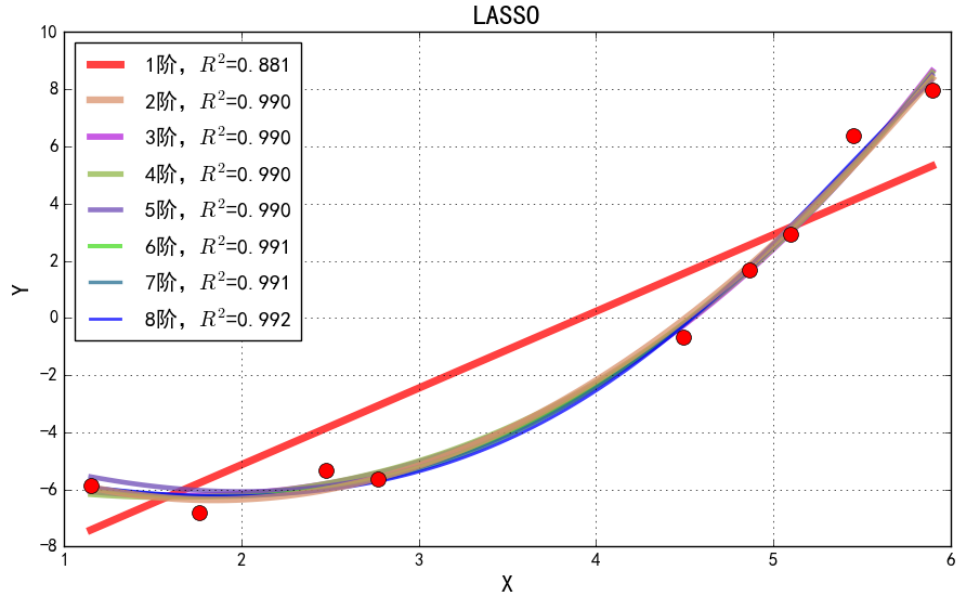
超参与过拟合

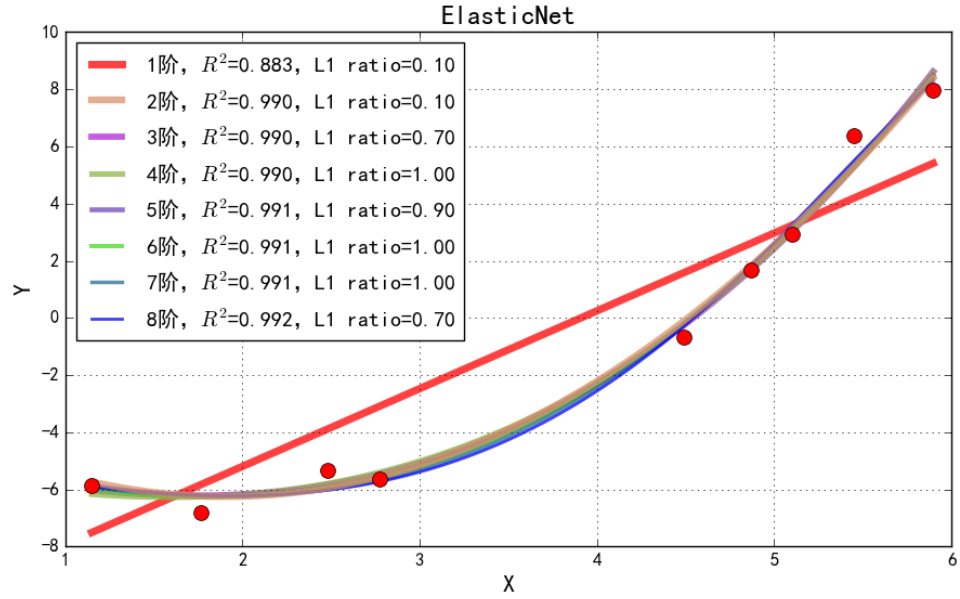

多项式曲线拟合
线性回归
1阶，R^2=0.896
2阶，R^2=0.990
3阶，R^2=0.991
4阶，R^2=0.991
5阶，R^2=0.997
6阶，R^2=0.997
7阶，R^2=1.000
8阶，R^2=1.000
Y
X
Ridge回归
1阶，R^2=0.891
2阶，R^2=0.990
3阶，R^2=0.990
4阶，R^2=0.987
5阶，R^2=0.974
6阶，R^2=0.997
7阶，R^2=0.997
8阶，R^2=0.996
Y
X

多项式曲线拟合比较


线性回归
1阶，R^2=0.896
2阶，R^2=0.990
3阶，R^2=0.991
4阶，R^2=0.991
5阶，R^2=0.997
6阶，R^2=0.997
7阶，R^2=1.000
8阶，R^2=1.000
X
Y


Ridge回归
1阶，R^2=0.891
2阶，R^2=0.990
3阶，R^2=0.990
4阶，R^2=0.987
5阶，R^2=0.974
6阶，R^2=0.997
7阶，R^2=0.997
8阶，R^2=0.996
X
Y


LASSO
1阶，R^2=0.881
2阶，R^2=0.990
3阶，R^2=0.990
4阶，R^2=0.990
5阶，R^2=0.990
6阶，R^2=0.991
7阶，R^2=0.991
8阶，R^2=0.992
X
Y


ElasticNet
1阶，R^2=0.883，L1 ratio=0.10
2阶，R^2=0.990，L1 ratio=0.10
3阶，R^2=0.990，L1 ratio=0.70
4阶，R^2=0.990，L1 ratio=1.00
5阶，R^2=0.991，L1 ratio=0.90
6阶，R^2=0.991，L1 ratio=1.00
7阶，R^2=0.991，L1 ratio=1.00
8阶，R^2=0.992，L1 ratio=0.70
X
Y

北京市区域犯罪率分析

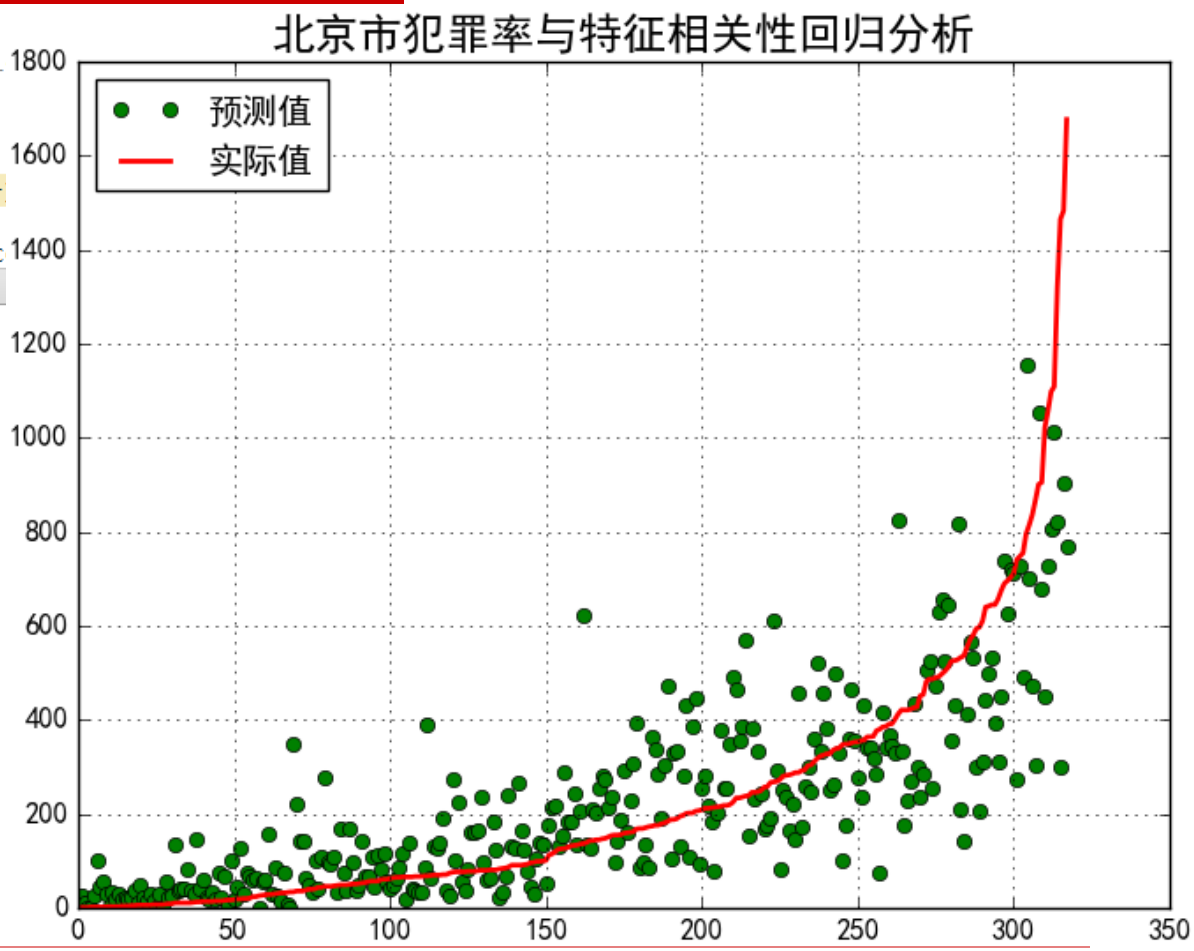
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	地区	盗窃案件数	批发和零售业数量	交通运输仓储邮政业数量	房地产业数量	住宿和餐饮业数量	卫生和社会工作数量	居民服务修理服务业数量	大型单位数量	中型单位数量	小微单位数量	金融业单位数量	液化石油气	能源合计	从业人员	销售费用	营业收入	营业税及附加	总产值	利润总额	人员支出
2	安定门街道办事处																				
3	安定镇																				
4	安贞街道办事处																				
5	奥运村街道办事处																				
6	八宝山街道办事处																				
7	八达岭镇																				
8	八角街道办事处																				
9	八里庄街道办事处(朝阳)																				
10	八里庄街道办事处（海淀）																				
11	白纸坊街道办事处																				
12	百泉街道办事处																				
13	百善镇																				
14	宝山镇																				
15	北房镇																				
16	北京经济技术开发区																				
17	北七家镇																				
18	北石槽镇																				
19	北太平庄街道办事处																				
20	北务镇																				
21	北下关街道办事处																				
22	北小营镇																				
23	北新桥街道办事处																				

北京市区域犯罪率分析

```
print 'model.alpha = \t', model.alpha_
# print 'model.l1_ratio = \t', model.l1_
print 'model.coef_ = \n', model.coef_
print 'model.predict(x) = \n', y_hat
print 'Acture = \n', y
print 'RMSE:\t', np.sqrt(np.mean((y_hat-
print 'R2:\t', model.score(x, y)
for theta, col in zip(model.coef_[1:], c
```

10.6.crim | 10.6.crim

```
RMSE:    162.964014931
R2: 0.626819207663
批发和零售业数量 128.651347274
交通运输仓储邮政业数量 60.9563623169
房地产业数量 20.864682258
住宿和餐饮业数量 12.3436281315
卫生和社会工作数量 10.6957507793
居民服务修理服务业数量 26.2299084155
中型单位数量 74.8997410393
从业人员 141.450826243
营业收入 5.9160254661
营业税及附加 20.429059238
利润总额 0.37297432004
人员支出 39.8048306311
地铁线路 4.24512985302
公交线路 44.8238997681
```



作业

- 推导Softmax回归的梯度公式。
- 参考给出的Logistic回归或线性回归代码，使用其他数据集做分类或预测实验。

我们在这里

- http://wenda.ChinaHadoop.cn
  - 视频/课程/社区
- 微博
  - @ChinaHadoop
  - @邹博_机器学习
- 微信公众号
  - 小象
  - 大数据分析挖掘

感谢大家！

恳请大家批评指正！