Statistics and Machine Learning

Classification II: LDA, QDA, and k-nearest neighbor

Week 8 03/08 — 03/12

Contents of Week 8

More models for tackling classification

- Review of Bayes rule
- Linear discriminant analysis (LDA)
- Quadratic discriminant analysis (QDA)
- K-nearest neighbor
- Lab session: predicting stock movement

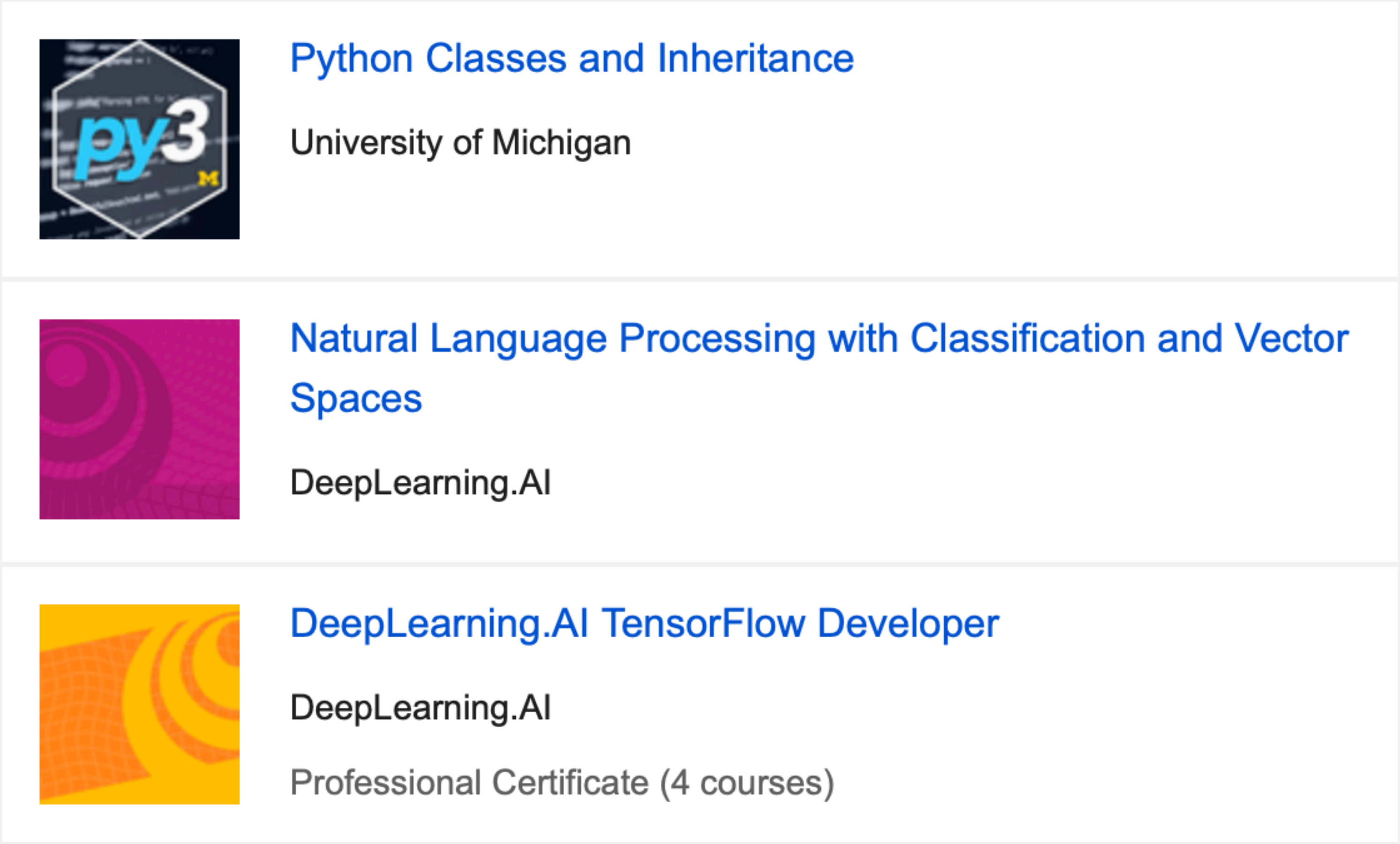
Administrative

Recommendation, spring break, mid-course survey

Administrative

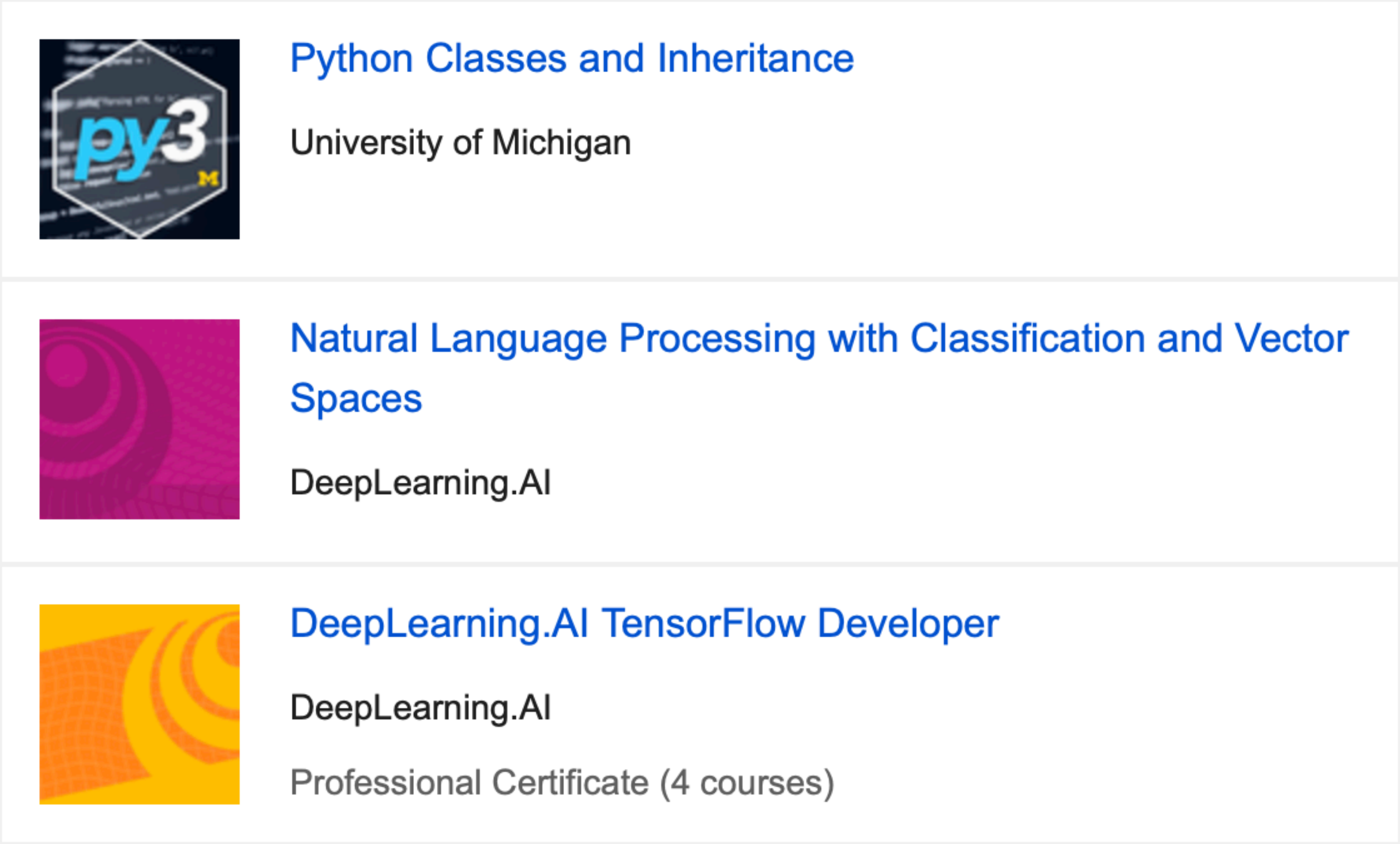
Recommendation, spring break, mid-course survey

- Coursera online course:



Administrative

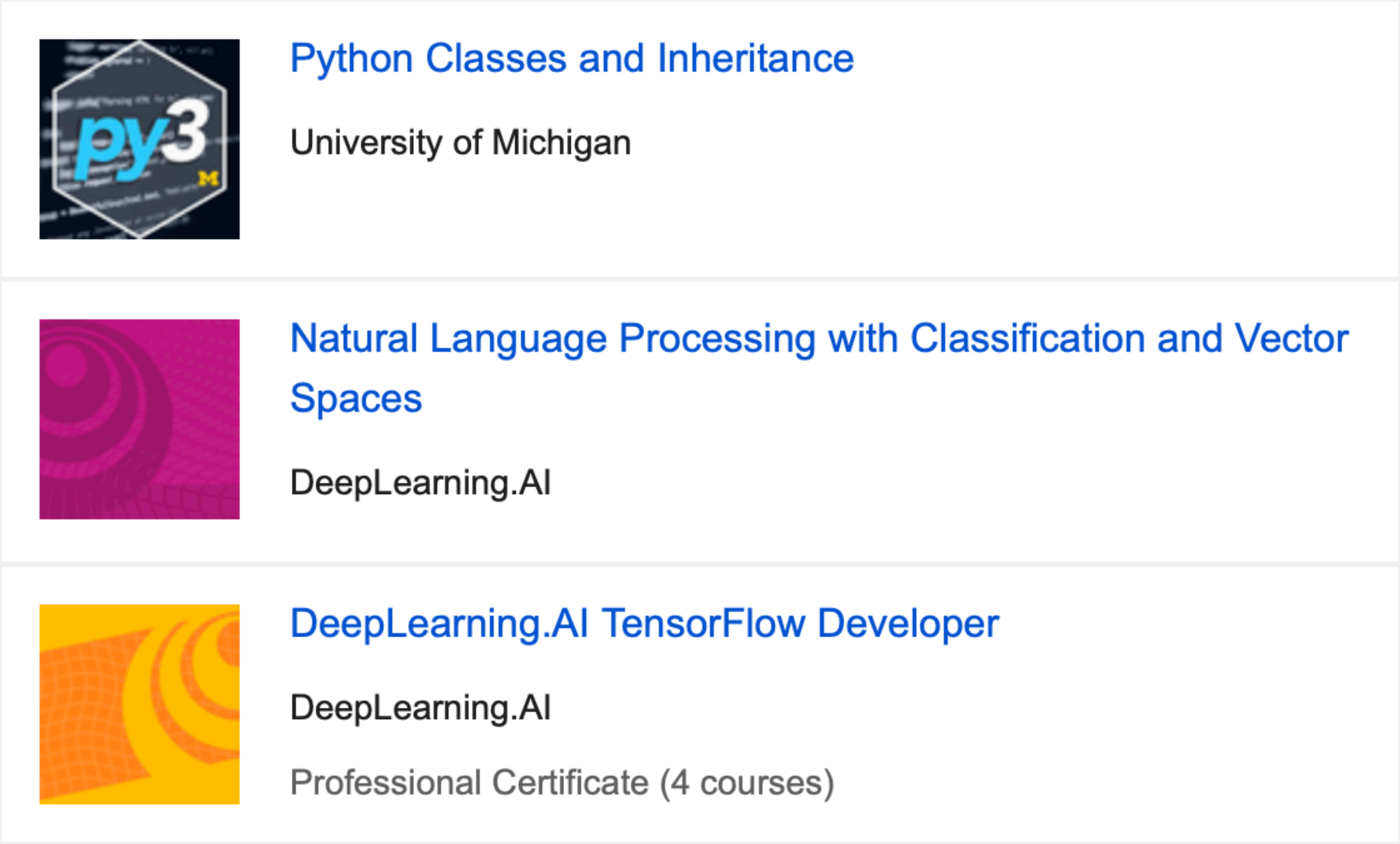
Recommendation, spring break, mid-course survey

- Coursera online course: 

- Spring break: week 9 (03/15 — 03/21). Midterm 03/24 8:00 pm

Administrative

Recommendation, spring break, mid-course survey



- Coursera online course:
- Spring break: week 9 (03/15 — 03/21). Midterm 03/24 8:00 pm
- Mid-course survey (1 point bonus)

Review on logistic regression and default data set

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, default

f<0.5: $\hat{y} = 0$, not default

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, default

f<0.5: $\hat{y} = 0$, not default

Loss function: cross entropy

$$L = \frac{1}{N}\sum_{n=1}^{N}[y \log f + (1-y)\log(1-f)]$$

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, **default**

f<0.5: $\hat{y} = 0$, **not default**

Loss function: cross entropy

$$L = \frac{1}{N}\sum_{n=1}^{N}[y\log f + (1-y)\log(1-f)]$$

Training accuracy:
(4799+55)/5000=97%

```
confusion_matrix(y_train, pred_train).T

array([[4799,  119],
       [  27,   55]])
```

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, default

f<0.5: $\hat{y} = 0$, not default

Loss function: cross entropy

$$L = \frac{1}{N}\sum_{n=1}^{N}[y\log f + (1-y)\log(1-f)]$$

Check week 7 lab code

Training accuracy:
(4799+55)/5000=97%

```
confusion_matrix(y_train, pred_train).T
```

```
array([[4799,  119],
       [  27,   55]])
```

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, default

f<0.5: $\hat{y} = 0$, not default

Loss function: cross entropy

$$L = \frac{1}{N}\sum_{n=1}^{N}[y \log f + (1-y)\log(1-f)]$$

Check week 7 lab code

Training accuracy:
(4799+55)/5000=97%

```
confusion_matrix(y_train, pred_train).T

array([[4799,  119],
       [  27,   55]])
```

Test accuracy:
(4813+53)/5000=97.3%

```
confusion_matrix(y_test, pred_test).T

array([[4813,  106],
       [  28,   53]])
```

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, default

f<0.5: $\hat{y} = 0$, not default

Loss function: cross entropy

$$L = \frac{1}{N}\sum_{n=1}^{N}[y\log f + (1-y)\log(1-f)]$$

Check week 7 lab code

Training accuracy:
(4799+55)/5000=97%

```
confusion_matrix(y_train, pred_train).T

array([[4799,  119],
       [  27,   55]])
```

Training accuracy (positive only):
55/(55+119)=31.6%

Test accuracy:
(4813+53)/5000=97.3%

```
confusion_matrix(y_test, pred_test).T

array([[4813,  106],
       [  28,   53]])
```

Review on logistic regression and default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function

$$f(x) = \frac{1}{1 + e^{-ax-b}}$$

X: balance

f>0.5: $\hat{y} = 1$, default

f<0.5: $\hat{y} = 0$, not default

Loss function: cross entropy

$$L = \frac{1}{N}\sum_{n=1}^{N}[y \log f + (1-y)\log(1-f)]$$

Check week 7 lab code

Training accuracy:
(4799+55)/5000=97%

```
confusion_matrix(y_train, pred_train).T
```

```
array([[4799,  119],
       [  27,   55]])
```

Training accuracy (positive only):
55/(55+119)=31.6%

Test accuracy:
(4813+53)/5000=97.3%

```
confusion_matrix(y_test, pred_test).T
```

```
array([[4813,  106],
       [  28,   53]])
```

Test accuracy (positive only):
53/(55+106)=33.3%

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function
$f(x) = 0$
X: balance
Always predict $\hat{y} = 0$, not default

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function
$f(x) = 0$
X: balance
Always predict $\hat{y} = 0$, not default

Loss function: cross entropy
Don't care

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function
$f(x) = 0$
X: balance
Always predict $\hat{y} = 0$, not default

Loss function: cross entropy
Don't care

Training accuracy:
(4826+0)/5000=96.5%

```
np.count_nonzero(y_train == 0)
```

```
4826
```

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function
$f(x) = 0$
X: balance
Always predict $\hat{y} = 0$, not default

Loss function: cross entropy
Don't care

Training accuracy:
(4826+0)/5000=96.5%

```
np.count_nonzero(y_train == 0)
```

```
4826
```

Training accuracy (positive only):
0%

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function
$f(x) = 0$
X: balance
Always predict $\hat{y} = 0$, not default

Loss function: cross entropy
Don't care

Training accuracy:
(4826+0)/5000=96.5%

```
np.count_nonzero(y_train == 0)
```

```
4826
```

Training accuracy (positive only):
0%

Test accuracy:
(4841+0)/5000=96.8%

```
np.count_nonzero(y_test == 0)
```

```
4841
```

Baseline model on default data set

Data set: default.csv
Default versus balance
10000 rows
9667 no, 333 yes

Model: logistic function
$f(x) = 0$
X: balance
Always predict $\hat{y} = 0$, not default

Loss function: cross entropy
Don't care

Training accuracy:
(4826+0)/5000=96.5%

```
np.count_nonzero(y_train == 0)
4826
```

Training accuracy (positive only):
0%

Test accuracy:
(4841+0)/5000=96.8%

```
np.count_nonzero(y_test == 0)
4841
```

Test accuracy (positive only):
0%

Review of Bayes rule

X: get a positive/negative report
Y: is/not a drug user

Sensitivity: $p(x=1|y=1) = 0.97$
Specificity: $p(x=0|y=0)=0.95$

Prior: $p(y=1)=0.005$

Question 1: $p(y=1|x=1)$

Question 2: $p(y=0|x=0)$

Review of Bayes rule

X: get a positive/negative report
Y: is/not a drug user

Sensitivity: $p(x=1|y=1) = 0.97$
Specificity: $p(x=0|y=0)=0.95$

Prior: $p(y=1)=0.005$

Question 1: $p(y=1|x=1)$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Question 2: $p(y=0|x=0)$

Review of Bayes rule

X: get a positive/negative report
Y: is/not a drug user

Sensitivity: p(x=1|y=1) = 0.97
Specificity: p(x=0|y=0)=0.95

Prior: p(y=1)=0.005

Question 1: p(y=1|x=1)

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)}$$

$$p(y=1 \mid x=1) = \frac{0.97 * 0.005}{0.97 * 0.005 + 0.05 * 0.995}$$

Question 2: p(y=0|x=0)

Review of Bayes rule

X: get a positive/negative report
Y: is/not a drug user

Sensitivity: $p(x=1|y=1) = 0.97$
Specificity: $p(x=0|y=0)=0.95$

Prior: $p(y=1)=0.005$

Question 1: $p(y=1|x=1)$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

$$p(y=1|x=1) = \frac{0.97 * 0.005}{0.97 * 0.005 + 0.05 * 0.995}$$

Question 2: $p(y=0|x=0)$

$$p(y=0|x=0) = \frac{0.95 * 0.995}{0.95 * 0.995 + 0.03 * 0.005}$$

Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Distribution of balance: p(x|y=1)
Distribution of balance: p(x=0|y=0)

Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Distribution of balance: $p(x|y=1)$
Distribution of balance: $p(x=0|y=0)$

Prior: $p(y=1) = 333/10000$

Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Distribution of balance: p(x|y=1)
Distribution of balance: p(x=0|y=0)

Prior: p(y=1) = 333/10000

$$p(x\,|\,y=0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-(x-\mu_0)^2/2\sigma_0^2}$$

Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Distribution of balance: p(x|y=1)
Distribution of balance: p(x=0|y=0)

Prior: p(y=1) = 333/10000

$$p(x\,|\,y=0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-(x-\mu_0)^2/2\sigma_0^2} \qquad p(x\,|\,y=1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2}$$

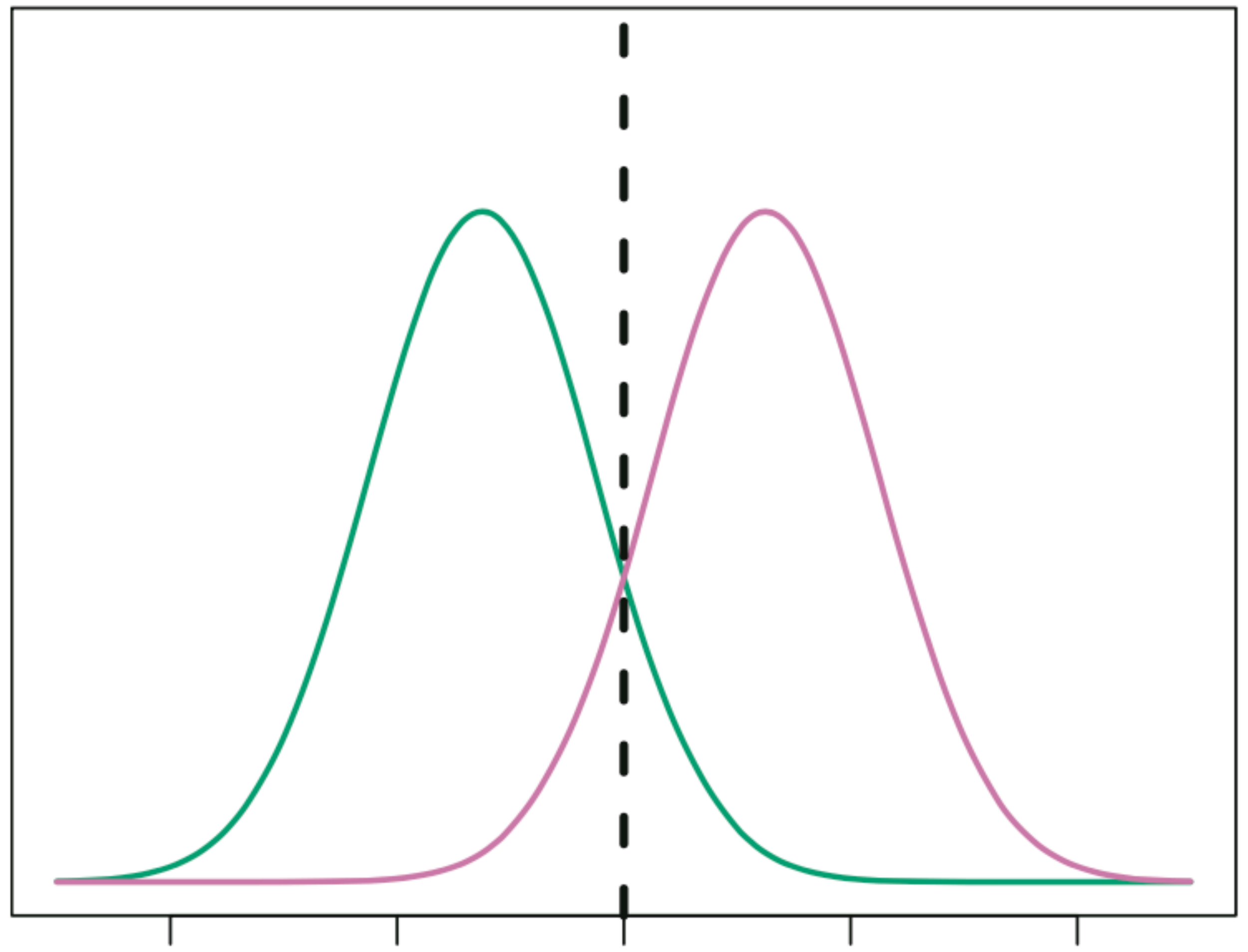
Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Distribution of balance: p(x|y=1)
Distribution of balance: p(x=0|y=0)

Prior: p(y=1) = 333/10000

$$p(x\,|\,y=0) = \frac{1}{\sqrt{2\pi}\sigma_0}e^{-(x-\mu_0)^2/2\sigma_0^2} \qquad p(x\,|\,y=1) = \frac{1}{\sqrt{2\pi}\sigma_1}e^{-(x-\mu_1)^2/2\sigma_1^2}$$



Bayes rule applied to default data

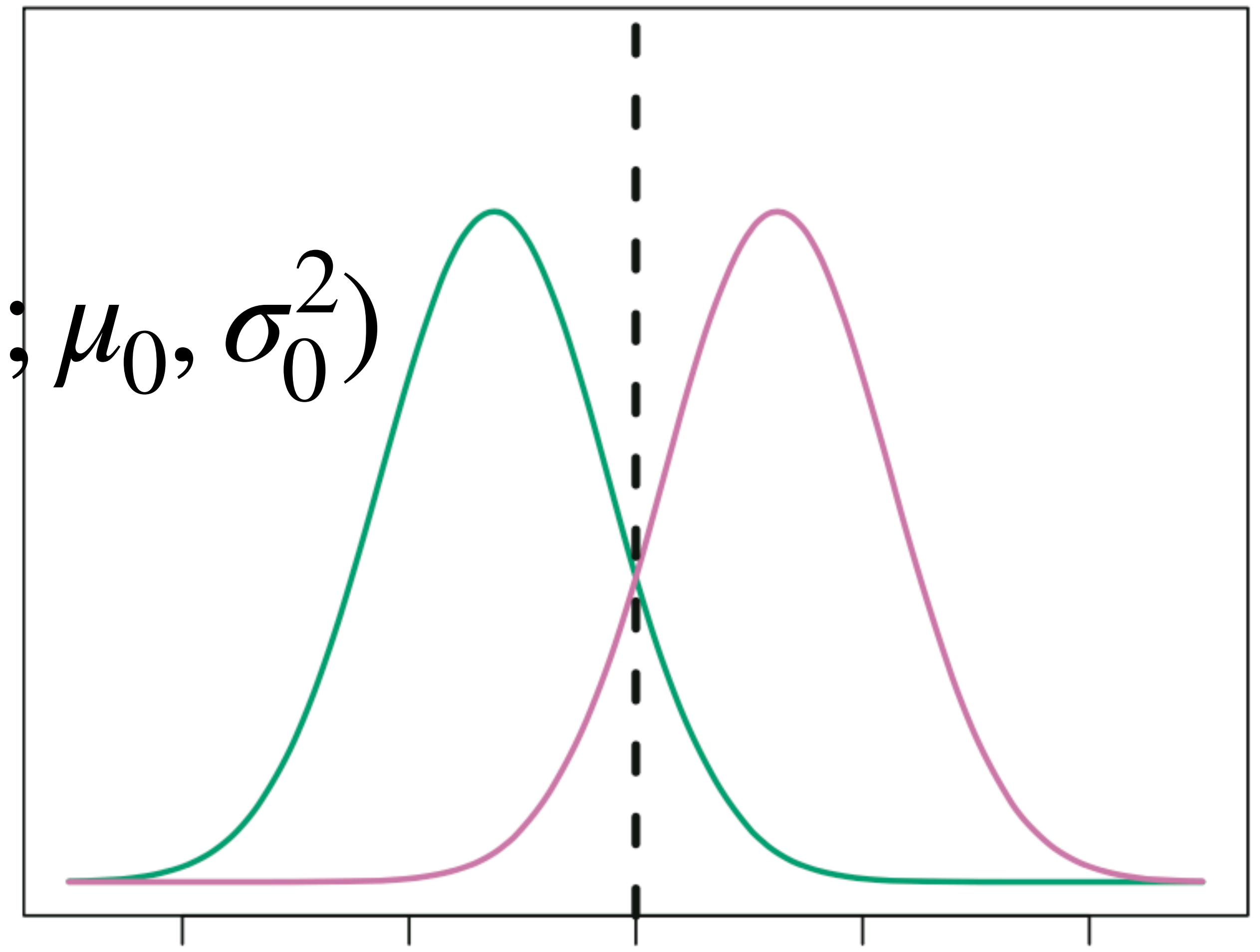
X: amount of balance
Y: default positive/negative

Distribution of balance: p(x|y=1)
Distribution of balance: p(x=0|y=0)

Prior: p(y=1) = 333/10000

$$p(x\,|\,y=0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-(x-\mu_0)^2/2\sigma_0^2} \qquad p(x\,|\,y=1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2}$$

$\mathbb{N}(x; \mu_0, \sigma_0^2)$



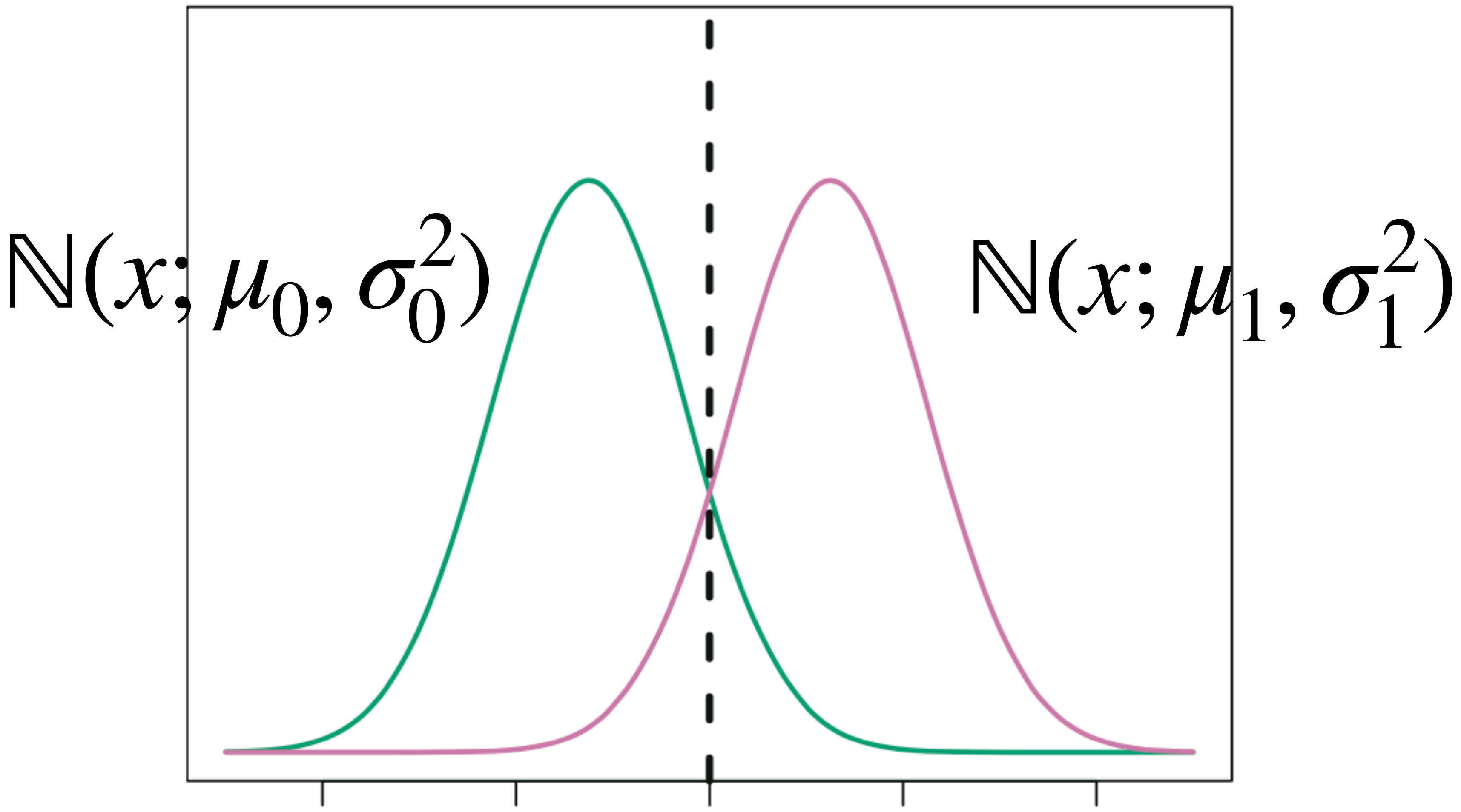
Bayes rule applied to default data

X: amount of balance
Y: default positive/negative

Distribution of balance: p(x|y=1)
Distribution of balance: p(x=0|y=0)

Prior: p(y=1) = 333/10000

$$p(x\,|\,y=0) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-(x-\mu_0)^2/2\sigma_0^2} \qquad p(x\,|\,y=1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-(x-\mu_1)^2/2\sigma_1^2}$$



Bayes rule applied to default data

The chance a person is default given x balance:

The chance a person is not default given x balance:

Bayes rule applied to default data

The chance a person is default given x balance:

$$p(y = 1 \mid x) \propto p(x \mid y = 1)p(y = 1) = \mathbb{N}(x; \mu_1, \sigma_1^2) * \pi_1$$

The chance a person is not default given x balance:

Bayes rule applied to default data

The chance a person is default given x balance:

$$\boxed{p(y=1\,|\,x) \propto p(x\,|\,y=1)p(y=1) = \mathbb{N}(x;\mu_1,\sigma_1^2) * \pi_1}$$

The chance a person is not default given x balance:

$$\boxed{p(y=0\,|\,x) \propto p(x\,|\,y=0)p(y=0) = \mathbb{N}(x;\mu_0,\sigma_0^2) * \pi_0}$$

Linear Discriminant Analysis (LDA)

Assumption: $\sigma_0 = \sigma_1$

$$\boxed{\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \qquad (4.13)}$$

Linear Discriminant Analysis (LDA)

Assumption: $\sigma_0 = \sigma_1$

$$\boxed{\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \qquad (4.13)}$$

If $\delta_0(x) > \delta_1(x)$**: prediction** $\hat{y} = 0$

Linear Discriminant Analysis (LDA)

Assumption: $\sigma_0 = \sigma_1$

$$\boxed{\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \qquad (4.13)}$$

If $\delta_0(x) > \delta_1(x)$**: prediction** $\hat{y} = 0$

If $\delta_0(x) < \delta_1(x)$**: prediction** $\hat{y} = 1$

Quadratic Discriminant Analysis (QDA)

Assumption: $\sigma_0 \neq \sigma_1$

Quadratic Discriminant Analysis (QDA)

Assumption: $\sigma_0 \neq \sigma_1$

$$\boxed{\delta_k(x) = -\frac{(x-\mu_k)^2}{2\sigma_k^2} - \log \sigma_k + \log \pi_k}$$

Quadratic Discriminant Analysis (QDA)

Assumption: $\sigma_0 \neq \sigma_1$

$$\boxed{\delta_k(x) = -\frac{(x-\mu_k)^2}{2\sigma_k^2} - \log \sigma_k + \log \pi_k}$$

If $\delta_0(x) > \delta_1(x)$**: prediction** $\hat{y} = 0$

Quadratic Discriminant Analysis (QDA)

Assumption: $\sigma_0 \neq \sigma_1$

$$\boxed{\delta_k(x) = -\frac{(x-\mu_k)^2}{2\sigma_k^2} - \log \sigma_k + \log \pi_k}$$

If $\delta_0(x) > \delta_1(x)$**: prediction** $\hat{y} = 0$

If $\delta_0(x) < \delta_1(x)$**: prediction** $\hat{y} = 1$

What are the parameters?

If we split the 10000 data into 5000 training and 5000 test:

μ_0**: the mean of balance of training data with y** $=$ **0**

σ_0^2**: variance of balance of training data with y** $=$ **0**

What are the parameters?

If we split the 10000 data into 5000 training and 5000 test:

μ_0**: the mean of balance of training data with y = 0**

μ_1**: the mean of balance of training data with y = 1**

σ_0^2**: variance of balance of training data with y = 0**

What are the parameters?

If we split the 10000 data into 5000 training and 5000 test:

μ_0: **the mean of balance of training data with y = 0**

μ_1: **the mean of balance of training data with y = 1**

σ_0^2: **variance of balance of training data with y = 0**

σ_1^2: **variance of balance of training data with y = 1**

What are the parameters?

If we split the 10000 data into 5000 training and 5000 test:

μ_0**: the mean of balance of training data with y = 0**

μ_1**: the mean of balance of training data with y = 1**

σ_0^2**: variance of balance of training data with y = 0**

σ_1^2**: variance of balance of training data with y = 1**

Quiz: if we treat the first 5000 row in default.csv as training subset, what are the values of the above four parameters?

K-nearest neighbor method

Please take notes

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	−1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

t=(0,0,0); ob1 =(0,3,0)

K-nearest neighbor method

Please take notes

d(t,ob1)	d(t,ob2)	d(t,ob3)	d(t,ob4)	d(t,ob5)	d(t,ob6)
$\sqrt{0^2+3^2+0^2}=3$	2	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{3}$

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	−1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

t=(0,0,0); ob1 =(0,3,0)

K-nearest neighbor method

Please take notes

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

t=(0,0,0); ob1 =(0,3,0)

d(t,ob1)	d(t,ob2)	d(t,ob3)	d(t,ob4)	d(t,ob5)	d(t,ob6)
$\sqrt{0^2+3^2+0^2}=3$	2	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{3}$

When k = 1,

K-nearest neighbor method

Please take notes

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

t=(0,0,0); ob1 =(0,3,0)

d(t,ob1)	d(t,ob2)	d(t,ob3)	d(t,ob4)	d(t,ob5)	d(t,ob6)
$\sqrt{0^2+3^2+0^2} = 3$	2	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{3}$

When k = 1,

Closest: ob.5 -> prediction = green

K-nearest neighbor method

Please take notes

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	−1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

t=(0,0,0); ob1 =(0,3,0)

d(t,ob1)	d(t,ob2)	d(t,ob3)	d(t,ob4)	d(t,ob5)	d(t,ob6)
$\sqrt{0^2+3^2+0^2} = 3$	2	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{3}$

When k = 1,

Closest: ob.5 -> prediction = green

When k = 3,

K-nearest neighbor method

Please take notes

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Obs.	X_1	X_2	X_3	Y
1	0	3	0	Red
2	2	0	0	Red
3	0	1	3	Red
4	0	1	2	Green
5	-1	0	1	Green
6	1	1	1	Red

Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

t=(0,0,0); ob1 =(0,3,0)

d(t,ob1)	d(t,ob2)	d(t,ob3)	d(t,ob4)	d(t,ob5)	d(t,ob6)
$\sqrt{0^2+3^2+0^2} = 3$	2	$\sqrt{10}$	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{3}$

$\boxed{\textbf{When k = 1,}}$

$\boxed{\textbf{Closest: ob.5 -> prediction = green}}$

$\boxed{\textbf{When k = 3,}}$

$\boxed{\textbf{Closest: ob. 5, 6, 2 —> prediction = (red*2+green*1)/3 = red}}$

Lab session

Stock market data, predicting movement for next day

```python
data = pd.read_csv('Smarket.csv')
```

```python
data.head(10)
```

	Unnamed: 0	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
0	1	2001	0.381	-0.192	-2.624	-1.055	5.010	1.1913	0.959	Up
1	2	2001	0.959	0.381	-0.192	-2.624	-1.055	1.2965	1.032	Up
2	3	2001	1.032	0.959	0.381	-0.192	-2.624	1.4112	-0.623	Down
3	4	2001	-0.623	1.032	0.959	0.381	-0.192	1.2760	0.614	Up
4	5	2001	0.614	-0.623	1.032	0.959	0.381	1.2057	0.213	Up
5	6	2001	0.213	0.614	-0.623	1.032	0.959	1.3491	1.392	Up
6	7	2001	1.392	0.213	0.614	-0.623	1.032	1.4450	-0.403	Down
7	8	2001	-0.403	1.392	0.213	0.614	-0.623	1.4078	0.027	Up
8	9	2001	0.027	-0.403	1.392	0.213	0.614	1.1640	1.303	Up
9	10	2001	1.303	0.027	-0.403	1.392	0.213	1.2326	0.287	Up

Using logistic regression, Lag1-5, volume, all data for training

```python
(507+144)/(507+144+458+141)
```

```
0.5208
```

Use 2001-2004 as training, 2005 as test

```python
confusion_matrix(y_test, pred_test).T
```

```
array([[48, 37],
       [93, 74]])
```

```python
(48+74)/(48+74+37+93)
```

```
0.48412698412698413
```

How to improve? LDA, QDA, k-nn.
See Lab Code