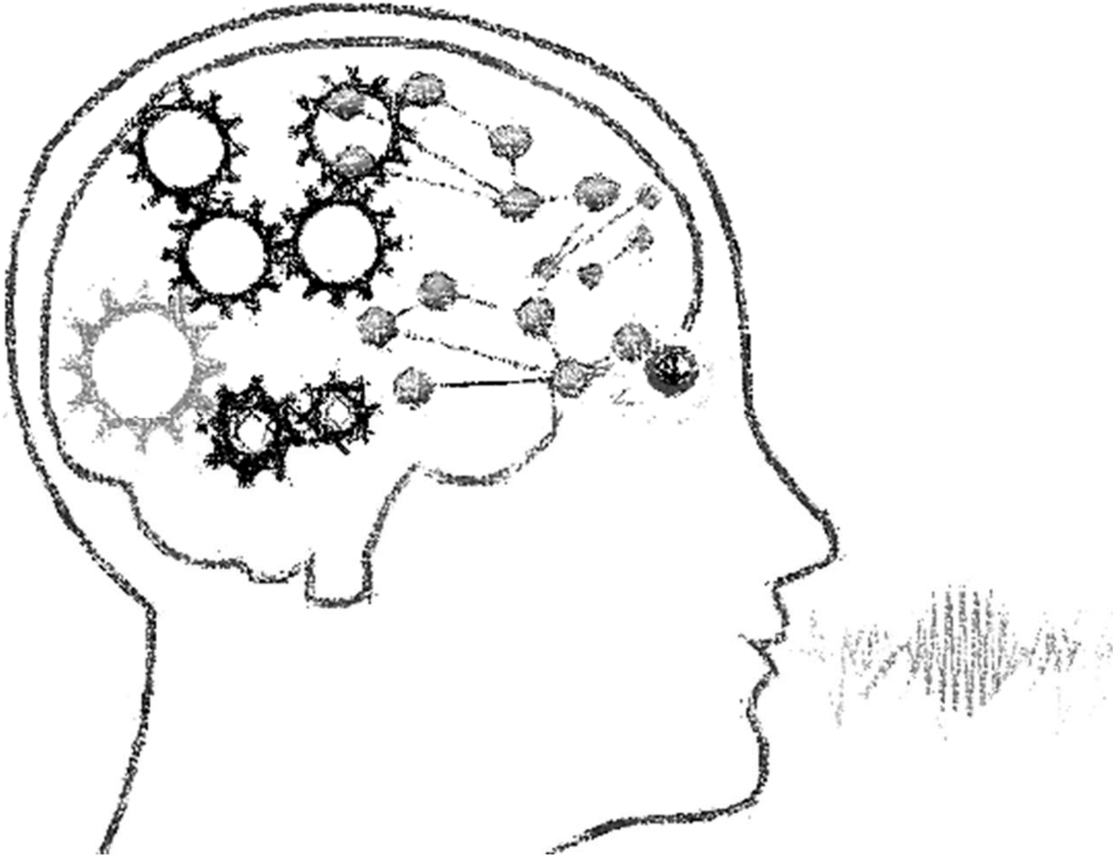
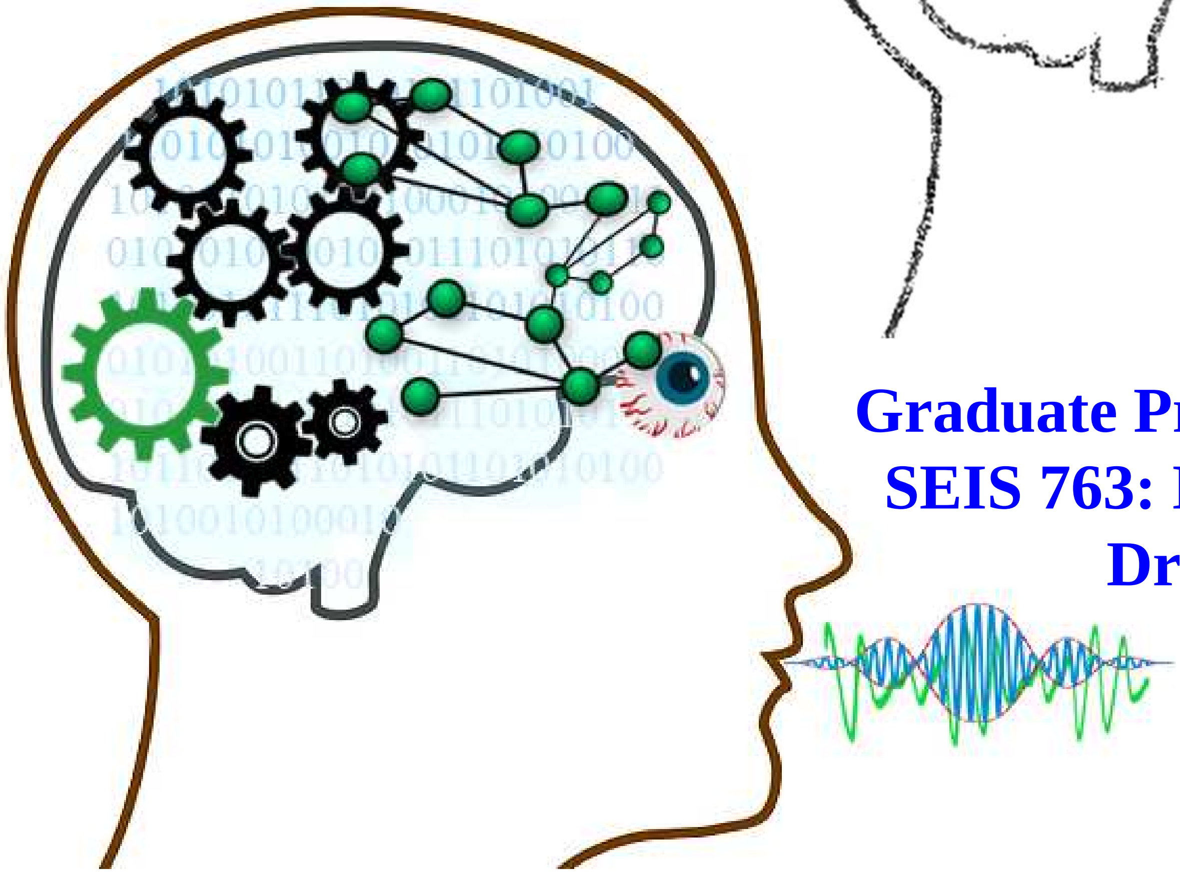
Classification Quality





Graduate Program in Software
SEIS 763: Machine Learning
Dr. Chih Lai

Outline

- How it works? Matlab / Python functions.
- Diagnoses / Visualization.
- **Prediction and Quality.**
- **Outliers.**
- Multiclasses Prediction.
- Nonlinearity / High Dimensionality / Visualization.
- Regularization.
- Cost Function.

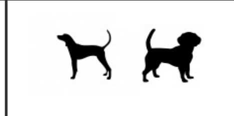
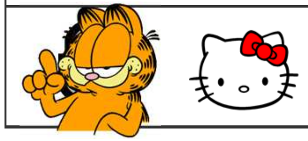
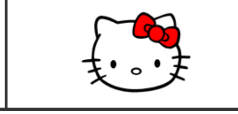
Confusion Matrix ➔ **Also Show Class Distribution**

a: **TP** (true positive) b: **FN** (false negative)

c: **FP** (false positive) d: **TN** (true negative)

	PREDICTED CLASS from DT		
ACTUAL CLASS in test data		Class=Yes	Class=No
	Class=Yes	a **(TP)**	b **(FN)**
	Class=No	c **(FP)**	d **(TN)**

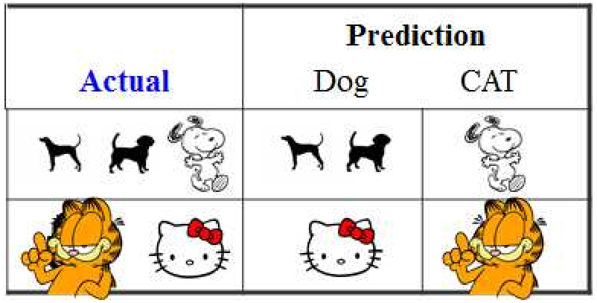
Example		Yes (210)	No (290)
class in test data	Yes (190)	**150**	40
	No (310)	60	**250**

Actual	**Prediction** Dog	CAT
		
		

- Consider a 2-class problem
  - All correct predictions are in the **diagonal** of the matrix
  - Errors are represented by non-zero values **outside** the diagonal

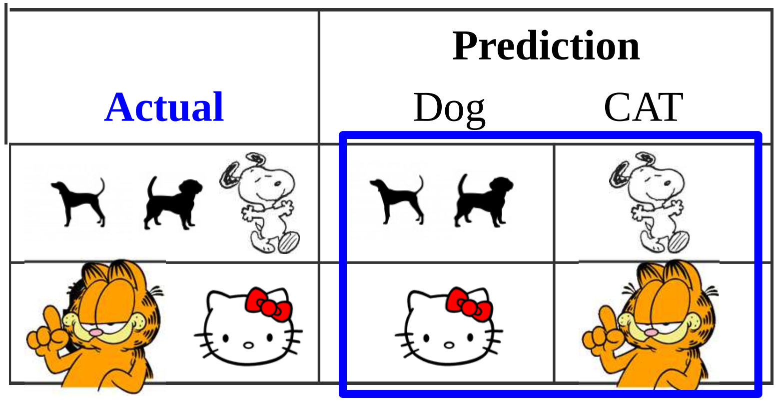
Measures Based on Confusion Matrix

	PREDICTED CLASS from DT	
		Class=Yes	Class=No
ACTUAL CLASS in test data	Class=Yes	(TP)	(FN)
	Class=No	(FP)	(TN)



Accuracy, Cost	$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
TP & FP rate, ROC Curve	$$\text{TP\_Rate} = \frac{TP}{TP + FN} \qquad \text{FP\_Rate} = \frac{FP}{TN + FP}$$
Recall, Precision, F1	$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{Precision} = \frac{TP}{TP + FP} \qquad \text{F1} = \frac{2 \times R \times P}{R + P}$$

Accuracy, Recall, Precision



- **Overall quality**

$$Accuracy_{dog,\ cat} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{3}{5} = 0.6$$

- Dog prediction

$$Recall = \frac{\#\ correct\ predictions}{\#\ records\ in\ the\ class} = \frac{TP=2}{TP+FN=(2+1)} = 0.66$$

$$Precision = \frac{\#\ correct\ predictions}{\#\ predictions\ in\ the\ class} = \frac{TP=2}{TP+FP=(2+1)} = 0.66$$

F1 = 0.66

$$F_1 = \frac{2 \times precison \times recall}{precison + recall}$$

Dog	PREDICTED CLASS	
	Dog	Cat
ACTUAL CLASS in test data Dog	TP	FN
Cat	FP	TN

- Cat prediction

$$Recall = \frac{\#\ correct\ predictions}{\#\ records\ in\ the\ class} = \frac{TP=1}{TP+FN=(1+1)} = 0.5$$

$$Precision = \frac{\#\ correct\ predictions}{\#\ predictions\ in\ the\ class} = \frac{TP=1}{TP+FP=(1+1)} = 0.5$$

F2 = 0.5

Cat	PREDICTED CLASS	
	Dog	Cat
ACTUAL CLASS in test data Dog	TN = 2	FP = 1
Cat	FN = 1	TP = 1

Cat	PREDICTED CLASS	
	Cat	Dog
ACTUAL CLASS in test data Cat	TP = 1	FN = 1
Dog	FP = 1	TN = 2

Accuracy, Recall, Precision **for "YES" Class Prediction**

- Accuracy = 0.8
- Recall = 0 / 20 = 0
- Precision = 0
- **F1** = NaN (i.e. 0)

Case 1		Yes (0)	No (100)
class in test data	Yes (20)	0	20
	No (80)	0	80

- Accuracy = 0.8
- Recall = 5 / 20 = 0.25
- Precision = 5 / 10 = 0.5
- **F1** = 0.33

Case 2		Yes (10)	No (90)
class in test data	Yes (20)	5	15
	No (80)	5	75

- Accuracy = 0.2
- Recall = 20 / 20 = 1
- Precision = 20 / 100 = 0.2
- **F1** = 0.33

Case 3		Yes (100)	No (0)
class in test data	Yes (20)	20	0
	No (80)	80	0

Accuracy, Recall, Precision for "NO" Class Prediction

- Accuracy = 0.8
- Recall = 80 / 80 = 1
- Precision = 80 / 100 = 0.8
- **F2** = 0.89

Case 1		Yes (0)	No (100)
class in test data	Yes (20)	0	20
	No (80)	0	80

- Accuracy = 0.8
- Recall = 75 / 80 = 0.94
- Precision = 75 / 90 = 0.83
- **F2** = 0.88

Case 2		Yes (10)	No (90)
class in test data	Yes (20)	5	15
	No (80)	5	75

- Accuracy = 0.2
- Recall = 0 / 80 = 0
- Precision = 0 / 0 = NaN
- **F2** = NaN (i.e. 0)

Case 3		Yes (100)	No (0)
class in test data	Yes (20)	20	0
	No (80)	80	0

Unbalance Data Is Everywhere

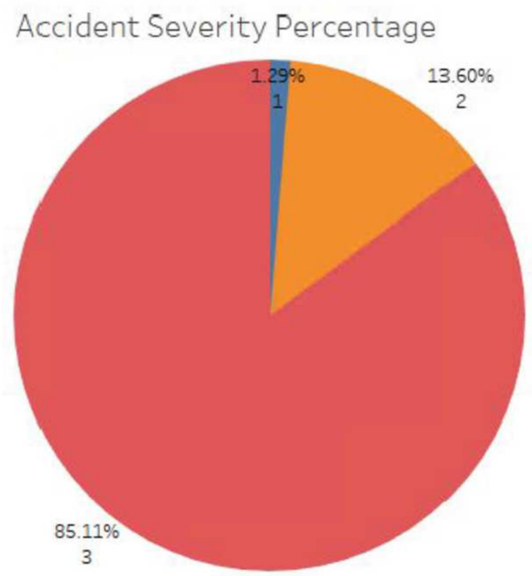
- # of alive patients = 2,219,192.
- # of deceased patients = 107,644   (0.0485).

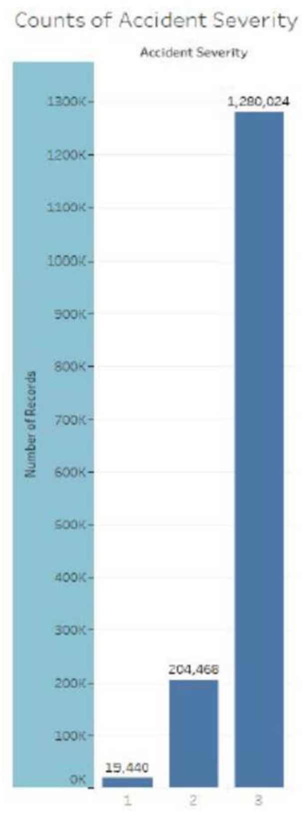
CFM for class-0 (deceased)

	0	**1**
0	7,485 **(TP)**	100,159 **(FN)**
1	2,232 **(FP)**	2,216,960 **(TN)**

Accuracy = 95.6%

- Studies show ***Naïve Bayes*** has the best performance on unbalanced data.





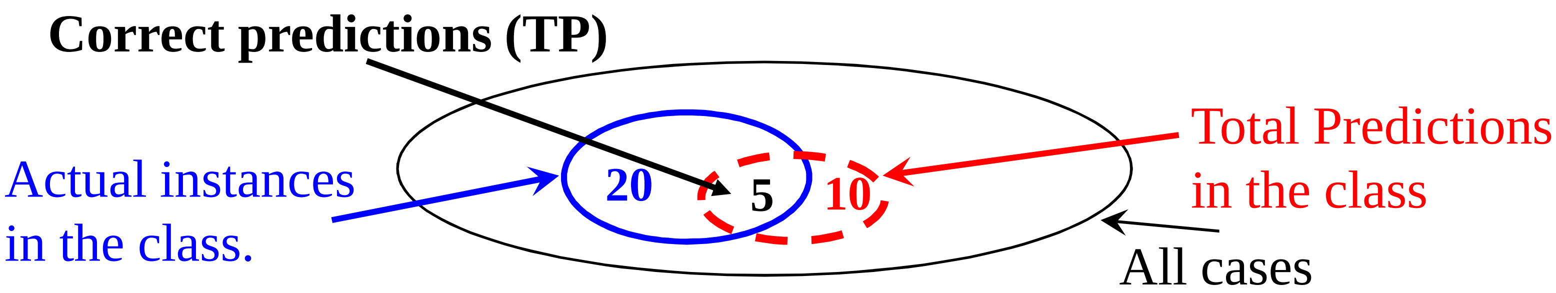
(Y) Class Prediction, Accuracy = 80%, Case 2

total=100	Predict (Y)	Predict (N)	
Actual (Y)	TP = 5	FN = 15	20
Actual (N)	FP = 5	TN = 75	80
	10	90	100

$$Recall = \frac{\#\, correct\ predictions}{\#\, records\ in\ the\ class} = \frac{TP=5}{TP+FN=(5+15)} = 0.25$$ (TP rate) **Low Recall!!!**

$$Precision = \frac{\#\, correct\ predictions}{\#\, predictions\ in\ the\ class} = \frac{TP=5}{TP+FP=(5+5)} = 0.5$$

$$F_1 = \frac{2 \times P \times R}{P+R} = \frac{2 \times 0.5 \times 0.25}{0.5+0.25} = 0.33$$



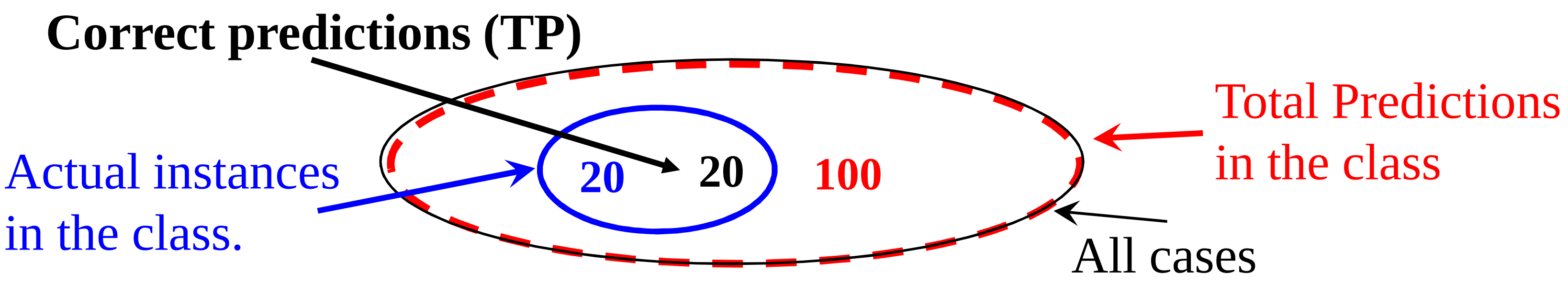
(Y) Class Prediction, Accuracy = 20%, Case 3

total=100	Predict (Y)	Predict (N)	
Actual (Y)	TP = 20	FN = 0	20
Actual (N)	FP = 80	TN = 0	80
	100	0	100

$$Recall = \frac{\#\, correct\ predictions}{\#\, records\ in\ the\ class} = \frac{TP{=}20}{TP+FN=(20+0)} = 1.0$$ (TP rate)

$$Precision = \frac{\#\, correct\ predictions}{\#\, predictions\ in\ the\ class} = \frac{TP{=}20}{TP+FP=(20+80)} = 0.2$$

$$F_1 = \frac{2\times P\times R}{P+R} = \frac{2\times 1\times 0.2}{1+0.2} = 0.33$$



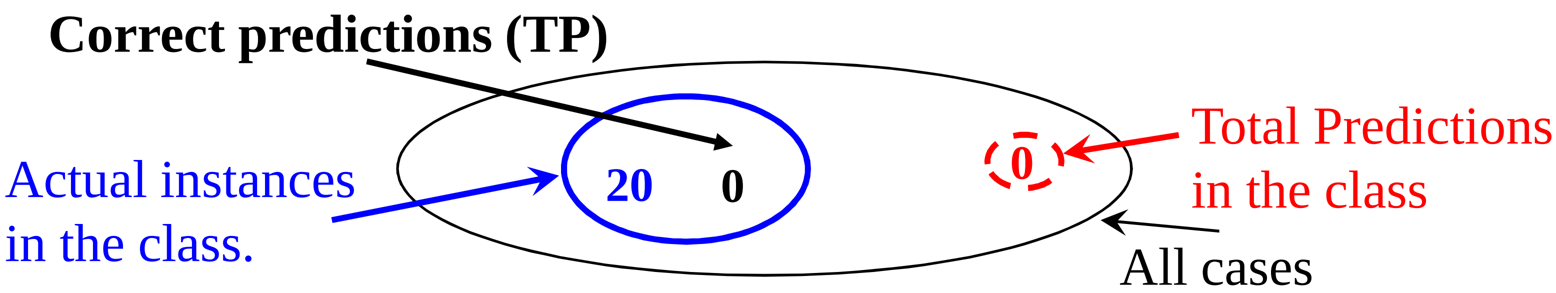
(Y) Class Prediction, Accuracy = 80%, Case 1

total=100	Predict (Y)	Predict (N)	
Actual (Y)	TP = 0	FN = 20	20
Actual (N)	FP = 0	TN = 80	80
	0	100	100

$$Recall = \frac{\#\ correct\ predictions}{\#\ records\ in\ the\ class} = \frac{TP=0}{TP+FN=(0+20)} = 0.0$$ (TP rate)

$$Precision = \frac{\#\ correct\ predictions}{\#\ predictions\ in\ the\ class} = \frac{TP=0}{TP+FP=(0+0)} = 0.0\ (???)$$

$$F_1 = \frac{2 \times precison \times recall}{precison + recall} = \frac{2 \times 0.0 \times 0.0}{0.0 + 0.0} = 0.0$$



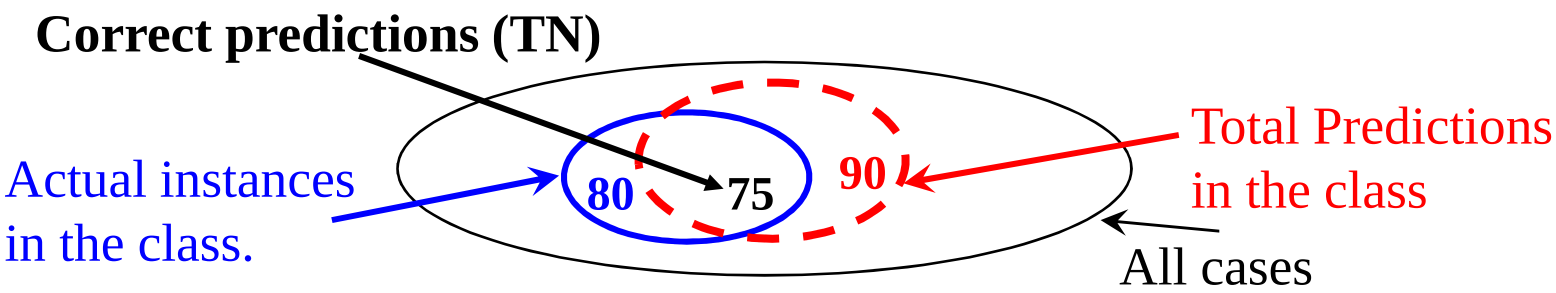
(N) Class Prediction, Accuracy = 80%, Case 2

total=100	Predict (Y)	Predict (N)	
Actual (Y)	TP = 5	FN = 15	20
Actual (N)	FP = 5	TN = 75	80
	10	90	100

$$Recall = \frac{\#\,correct\ predictions}{\#\,records\ in\ the\ class} = \frac{TN{=}75}{FP{+}TN{=}(5{+}75)} = \mathbf{0.94} \qquad \text{(TP rate)}$$

$$Precision = \frac{\#\,correct\ predictions}{\#\,predictions\ in\ the\ class} = \frac{TN{=}75}{FN{+}TN{=}(15{+}75)} = \mathbf{0.83}$$

$$F_1 = \frac{2\times P\times R}{P+R} = \frac{2\times 0.94\times 0.83}{0.94+0.83} = \mathbf{0.88}$$



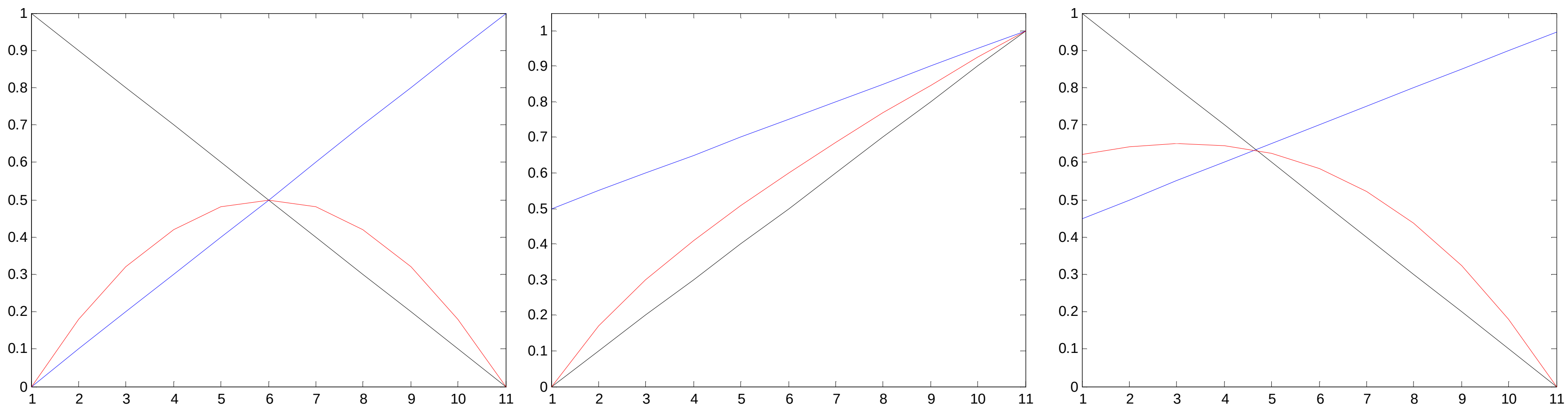
F Score– Combining Precision & Recall

- F Score
  - **F1**
  - **F2**

$$F = \frac{2 \times precison \times recall}{precison + recall}$$

total=100	Predict (Y)	Predict (N)	
Actual (Y)	TP = 20	FN = 0	20
Actual (N)	FP = 80	TN = 0	80
	100	0	100

Precision Recall F1

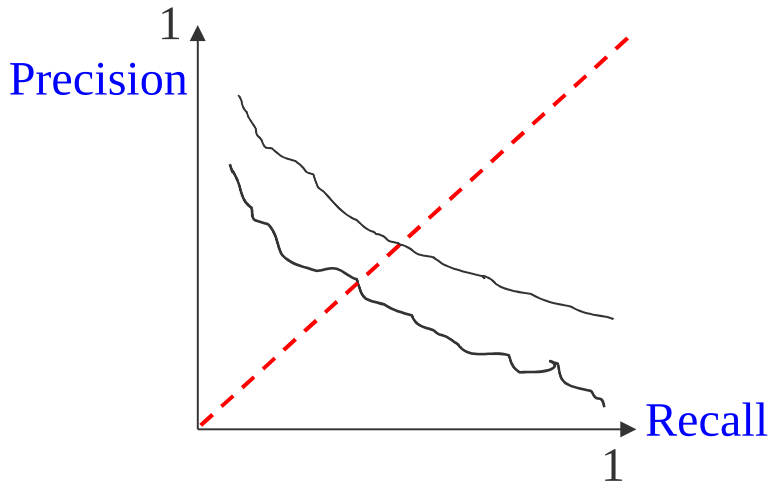


Trade Off and Sublinear Relationships

- Usually trade-off between these two measures

$$\textbf{Recall}: \frac{\text{\# of correct predictions} \uparrow}{\text{\# of instances of the class}} \qquad \textbf{Precision}: \frac{\text{\# of correct predictions} \uparrow}{\text{\# of predictions of the class} \uparrow}$$

Recall ↑   Precision ↓



- Other names for recall and precision
  - http://en.wikipedia.org/wiki/Sensitivity_and_specificity

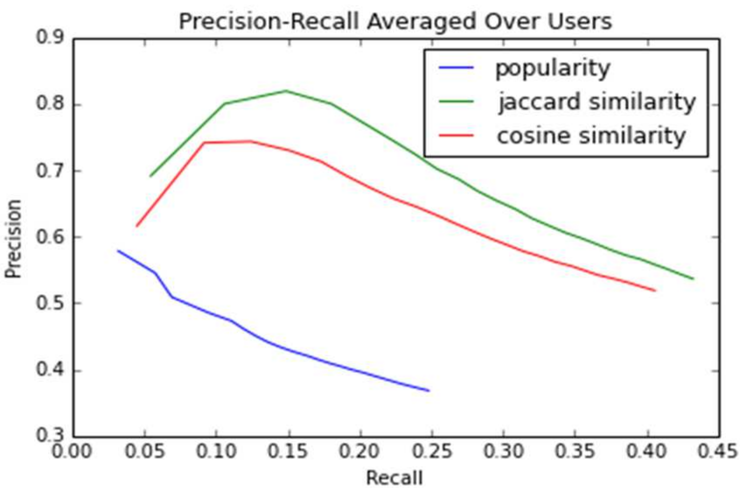
Movie Recommendation

- Similarity– measures the distance between users that have rated the same item
- Simple popularity– counts # of times movies being rated, & suggest popular ones

100 %

Results | Messages

	UserID	MovieID	Ratings	Occupation	Age	Gender	zipcode	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	Drama	Fantasy	FilmNoir	Horror	Romance	Musical	SciFi	Thriller	Unknown	War	Movie
1	82	1134	2	programmer	50	M	22902	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Get o
2	13	836	2	educator	47	M	29206	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	Nincto
3	13	272	4	educator	47	M	29206	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	Good
4	244	756	2	technician	28	M	80525	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	Father
5	305	427	5	programmer	23	M	94086	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	To Kill
6	95	787	2	administrator	31	M	10707	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	Rooms
7	43	14	2	librarian	29	F	20854	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	Positr
8	299	955	4	doctor	29	M	63108	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	Before
9	57	419	3	none	16	M	84010	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	Mary F
10	84	405	3	executive	32	M	55369	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	Mexico
11	269	504	4	librarian	31	F	43201	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	Bonnie
12	299	111	3	doctor	29	M	63108	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	Truth
13	194	466	4	administrator	38	M	2154	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Red R
14	160	135	4	programmer	27	M	66215	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	1	2001
15	99	268	3	student	20	M	63129	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	Chasr
16	10	486	4	lawyer	53	M	90703	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	Sabrin
17	259	117	4	student	21	M	48823	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Rock
18	85	427	3	educator	51	M	20003	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	To Kill
19	303	919	4	student	19	M	14853	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	City of
20	213	273	5	executive	33	M	55345	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	Heat /



2014 DM team

Adnan Al Alawiyat,

Lobabah Al Alawiyat,

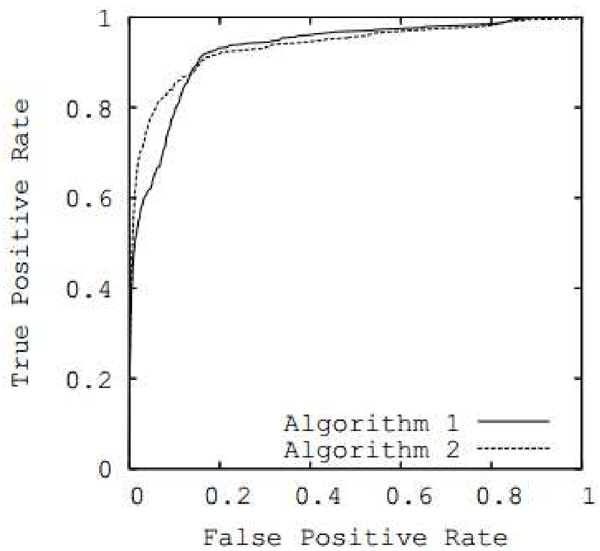
Grishma Iama,

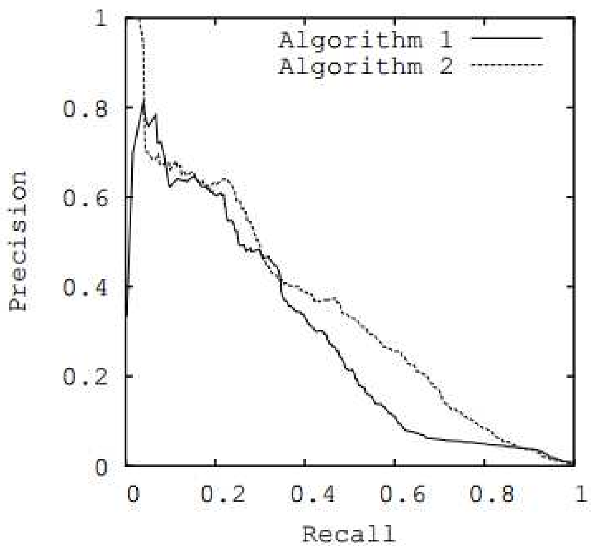
David Shiell,

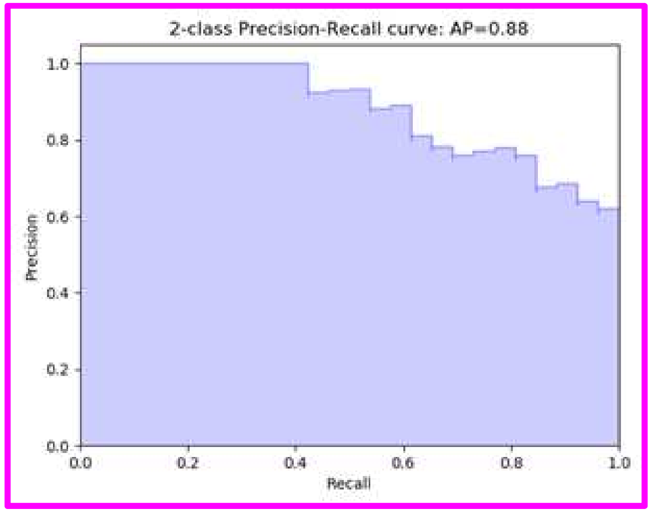
Russell Shurts

ROC AUC vs. PRC AUC

- Differences between the **ROC AUC** & **PR AUC**
  - http://www.chioka.in/differences-between-roc-auc-and-pr-auc/
  - http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html
  - http://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html







http://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html

True Positive (TP) & False Positive (FP) Rates

Confusion Matrix

	PREDICTED CLASS from DT		
ACTUAL CLASS in test data		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

A		Yes (210)	No (290)
class in test data	Yes (190)	150	40
	No (310)	60	250

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

- Total 500 test cases– 190 of Class **Yes**, 310 of Class **No**
- Prediction– 210 of Class **Yes**, 290 of Class **No**
- Accuracy = (150 + 250) / 500 = 80%
- **TP_% = 150 / 190 = 0.79** **FP_% = 60 / 310 = 0.19**
  - How good is the model making right "**guess**"?

Comparing Prediction Quality of ***EACH*** Class

- CFM tells us if our dataset has class-balancing issue.
- So, please always include CFM.

Training Dataset		Test Dataset
Precision Recall F	**Target Class**	**Precision Recall F**
0.9967 0.9957 **0.9962**	**1** **(1234)**	0.9955 0.9947 **0.9951**
0.9986 0.9989 **0.9988**	**2** **(2134)**	0.9983 0.9988 **0.9985**
0.9937 0.9953 **0.9945**	**3** **(3124)**	0.9923 0.9929 **0.9926**
0.9993 0.9991 **0.9992**	**4** **(4123)**	0.9992 0.9992 **0.9992**

Training Dataset			**Test Dataset**	
CFM	**Precision Recall F**	**Target Class**	**Precision Recall F**	**CFM**
104015 0 366 84 0 81030 83 5 280 52 71675 10 61 58 9 142272	0.9967 0.9957 **0.9962**	**1** **(1234)**	0.9955 0.9947 **0.9951**	26051 0 116 22 0 20206 20 5 104 22 17818 2 14 13 3 35604
81030 0 83 5 0 104015 366 84 52 280 71675 10 58 61 9 142272	0.9986 0.9989 **0.9988**	**2** **(2134)**	0.9983 0.9988 **0.9985**	20206 0 20 5 0 26051 116 22 22 104 17818 2 13 14 3 35604
71675 280 52 10 366 104015 0 84 83 0 81030 5 9 61 58 142272	0.9937 0.9953 **0.9945**	**3** **(3124)**	0.9923 0.9929 **0.9926**	17818 104 22 2 116 26051 0 22 20 0 20206 5 3 14 13 35604
142272 61 58 9 84 104015 0 366 5 0 81030 83 10 280 52 71675	0.9993 0.9991 **0.9992**	**4** **(4123)**	0.9992 0.9992 **0.9992**	35604 14 13 3 22 26051 0 116 5 0 20206 20 2 104 22 17818

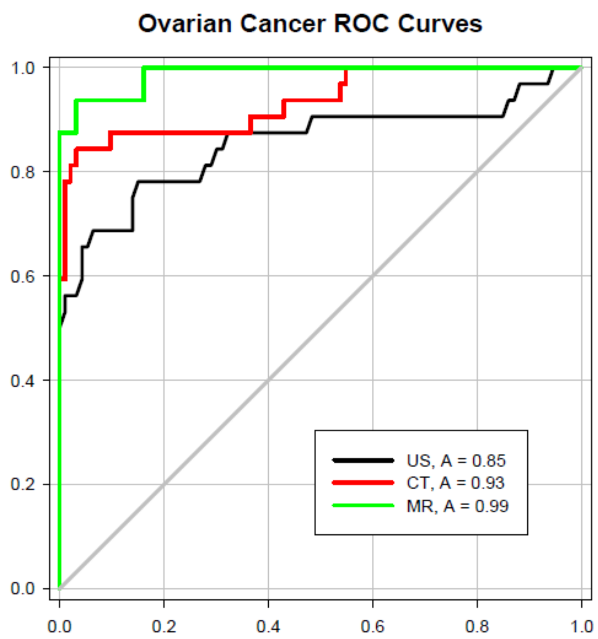
Satillite Image Classification

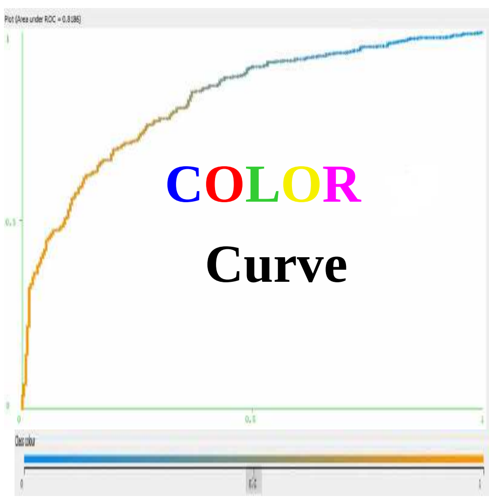
MDL-01, 2018 spring

Erin Jacot
Jared Gilbert
Khaled Mahmud
Laura Nicla
Lydia Xu

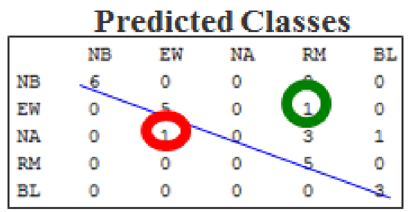
Visualizing Performance of Multiple Models

- Receiver Operating Characteristic (***ROC curve***)
  - Plot TP rate vs FP rate based on **prediction scores / quality**.
  - ***ROC*** curve closer to the upper left corner the better, WHY??
  - Area under curve (***AUC***) to compare performance of different models.
  - PCP plot.





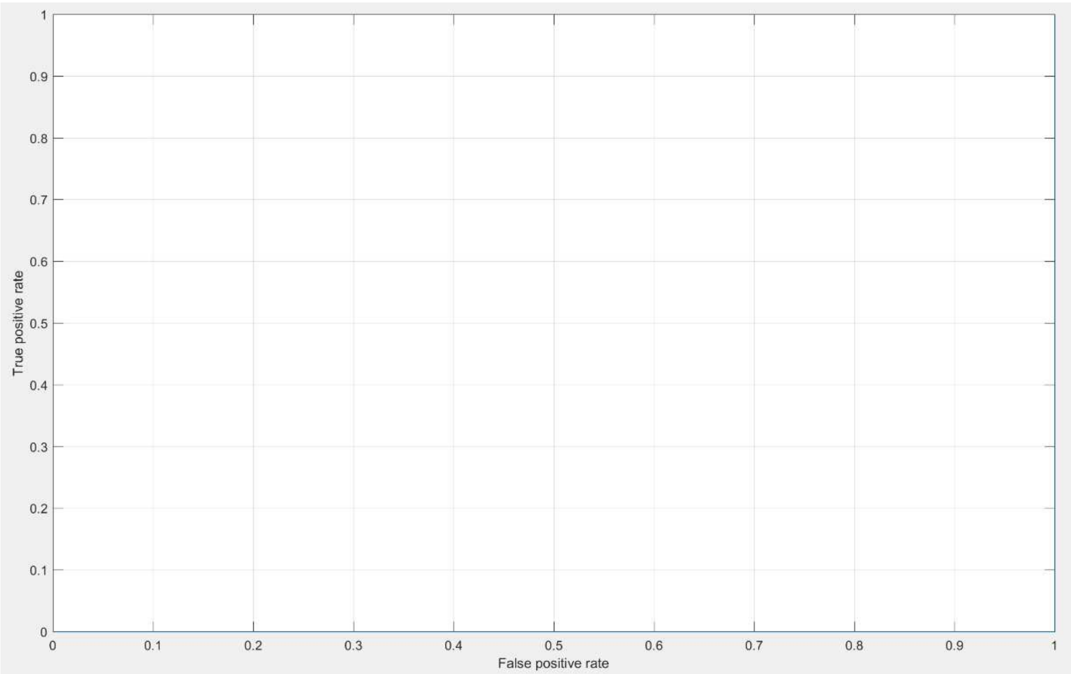
ROC Curve



	NB	EW	NA	RM	BL
NB	6	0	0	0	0
EW	0	5	0	1	0
NA	0	1	0	3	1
RM	0	0	0	5	0
BL	0	0	0	0	3

- Plot ROC curve & compute **AUC** (***Area Under Curve***) value
  - Larger AUC the better the model for the class

- Vary the *classification threshold* between 0..1 to generate an ***ROC curve***
  - ROC curve plots relationships using different THs and resulting TP and FP rates
  - If we keep reducing TH below 0.7, FP will increase
  - If we keep increasing TH above 0.57, TP keeps decreasing

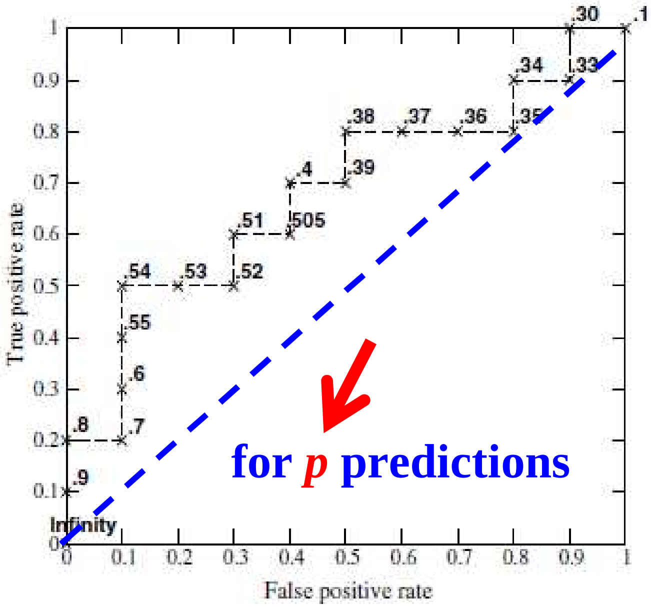


Score ≥	TP	FP	TPR/FPR
0.9 (M)	1	0	0.2 / 0
0.8 (M)	2	0	0.4 / 0
0.7 (F)	2	1	0.4 / 0.25
0.65 (M)	3	1	0.6 / 0.25
0.59 (M)	4	1	0.8 / 0.25
0.57 (F)	4	2	0.8 / 0.5
0.54 (M)	5	2	1 / 0.5
0.53 (F)	5	3	1 / 0.75
0.52 (F)	5	4	1 / 1

Build & Interpret ROC Curve, for "p" Class Prediction

Inst#	True Class	Score	Inst#	True Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

https://ccrma.stanford.edu/workshops/mir2009/references/ROCintro.pdf



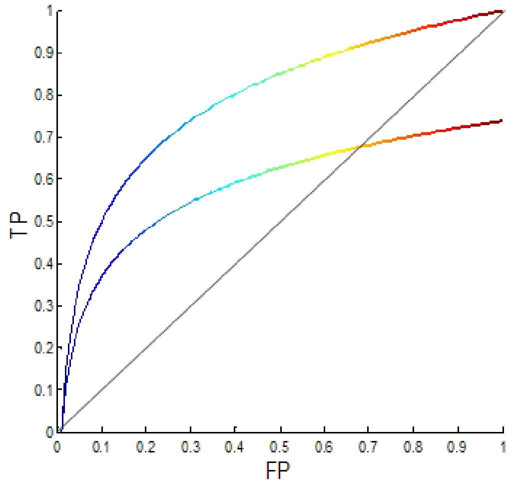
Score ≥	TP(TPR)	FP(FPR)
0.9	1 (0.1)	0 (0.0)
0.8	2 (0.2)	0 (0.0)
0.7	2 (0.2)	1 (0.1)
0.6	3 (0.3)	1 (0.1)
0.55	4 (0.4)	1 (0.1)
0.54	5 (0.5)	1 (0.1)
0.53	5 (0.5)	2 (0.2)
0.52	5 (0.5)	3 (0.3)
0.51	6 (0.6)	3 (0.3)

$$\text{TP\_Rate} = \frac{TP}{TP+FN} = \frac{TP}{\text{all positive cases}}$$

$$\text{FP\_Rate} = \frac{FP}{TN+FP} = \frac{FP}{\text{all negative cases}}$$

0.7	p	n
p	2 (TP)	(FN)
n	1 (FP)	(TN)

0.52	p	n
p	5 (TP)	(FN)
n	3 (FP)	(TN)

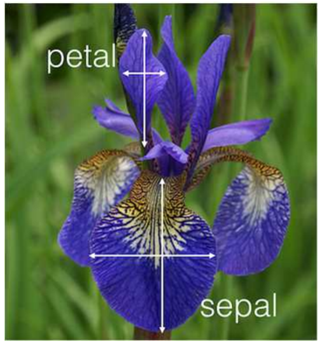


TH↓ ➔ FP↑

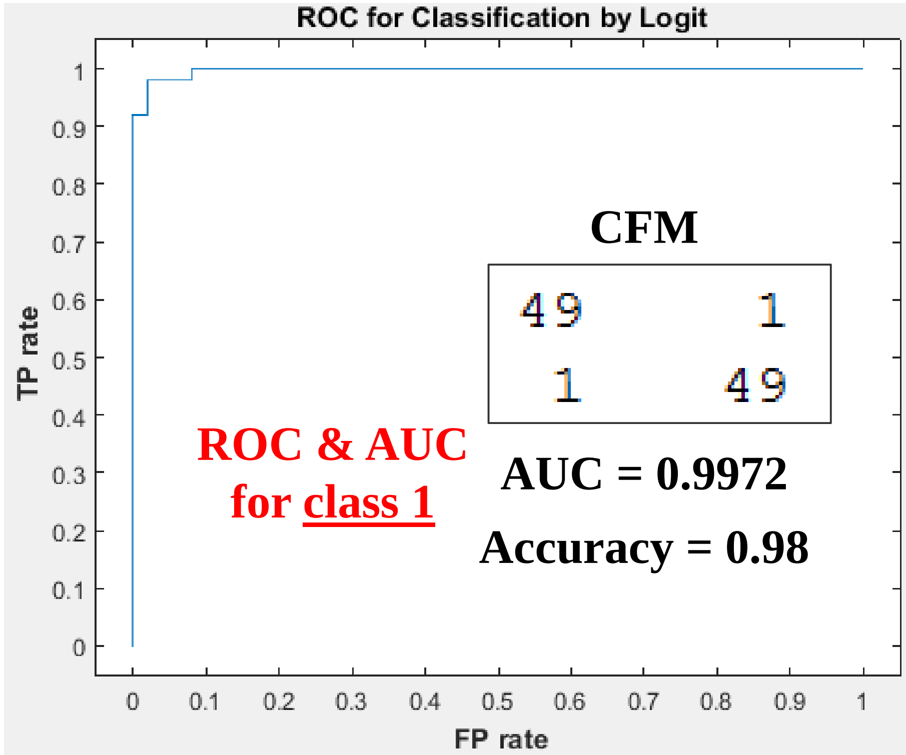
TH↑ ➔ FN↑

Prediction Accuracy and Confusion Matrix– Iris Example

```
load fisheriris
X = meas(51:end,:);          % 100×4
% Next, create 100×1    BINARY class
Y = double(strcmp('versicolor',species(51:end)));
mdl = fitglm(X, Y, 'distr', 'binomial', 'link', 'logit')

scores = predict(mdl, X);
PredictedClasses = double(scores >= 0.5);
CFM = confusionmat(Y,  PredictedClasses)

[xpos, ypos, T, AUC0] = perfcurve(Y, 1-scores, 0);
figure,     plot(xpos, ypos)
xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('\bf FP rate'),  ylabel('\bf TP rate')
title('\bf ROC for class 0 by Logit')

[xpos, ypos, T, AUC1] = perfcurve(Y, scores, 1);
figure,     plot(xpos, ypos)
xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('\bf FP rate'),  ylabel('\bf TP rate')
title('\bf ROC for class 1 by Logit')
```

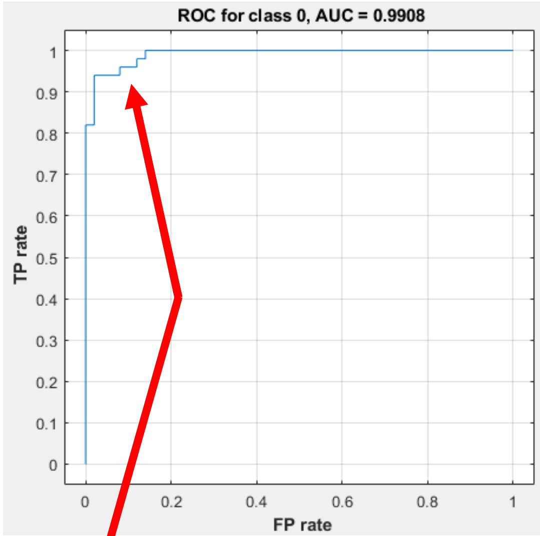


http://sebastianraschka.com/Articles/2014_python_lda.html



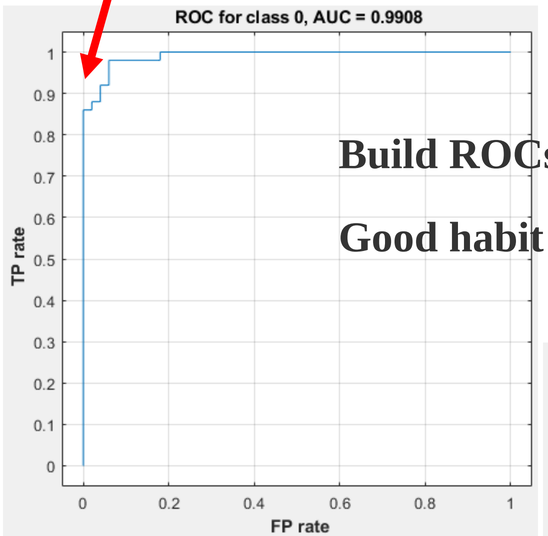
ROC Curve for **EACH** Binary Classes??? Why??? Always The Same???

[xpos, ypos, T, **AUC0**] = **perfcurve**(Y, **1-scores**, **0**);
figure,   plot(xpos, ypos)
xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('\bf FP rate'), ylabel('\bf TP rate')
title('\bf ROC for class **0** by Logit')




```
CFM =

      50     0
       1    49
```

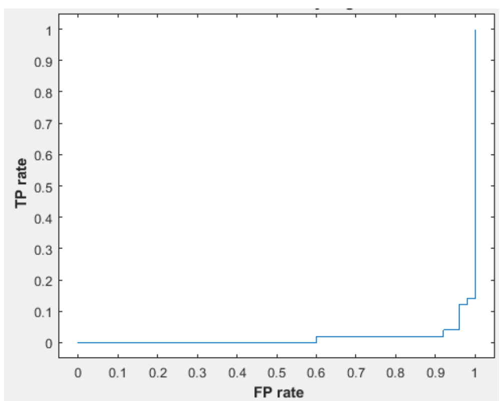
[xpos, ypos, T, **AUC1**] = **perfcurve**(Y, **scores**, **1**);
figure,   plot(xpos, ypos)
xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('\bf FP rate'), ylabel('\bf TP rate')
title('\bf ROC for class **1** by Logit')



Build ROCs for EACH class

Good habit for multi-classes

[xpos, ypos, T, **AUC1**] = **perfcurve**(Y, **scores**, **0**); **% if you use 0 here, then**



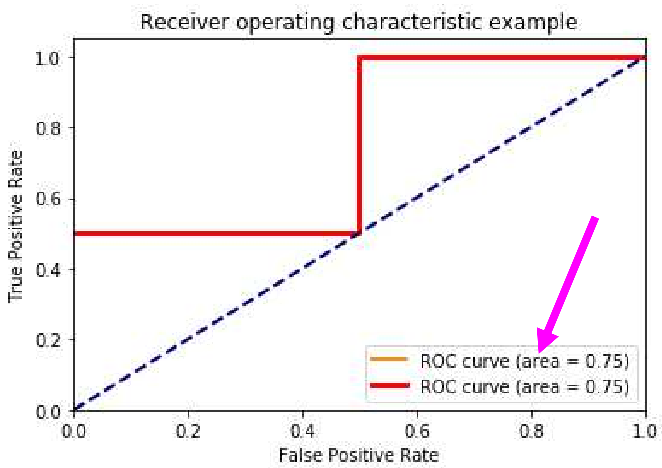
Python sklearn ROC, AUC

```
import numpy as np
from sklearn.metrics import roc_curve, auc

y = np.array([[1, 1, 0, 0], [0, 0, 1, 1]])
#a=np.array([0.1, 0.4, 0.35, 0.8])
scores=np.zeros( (2,4) )
scores[0, :] = 1-np.array([0.1, 0.4, 0.35, 0.8])
scores[1, :] = [0.1, 0.4, 0.35, 0.8]
#print(y)
print(scores)

[[ 0.9   0.6   0.65  0.2 ]
 [ 0.1   0.4   0.35  0.8 ]]
```

```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

plt.figure()
lw = 2
plt.plot(fpr[0], tpr[0], color='darkorange',
        lw=lw, label='ROC curve (area = %0.2f)' % roc_auc[0])
plt.plot(fpr[1], tpr[1], color='red',
        lw=3, label='ROC curve (area = %0.2f)' % roc_auc[1])
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0]); plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```

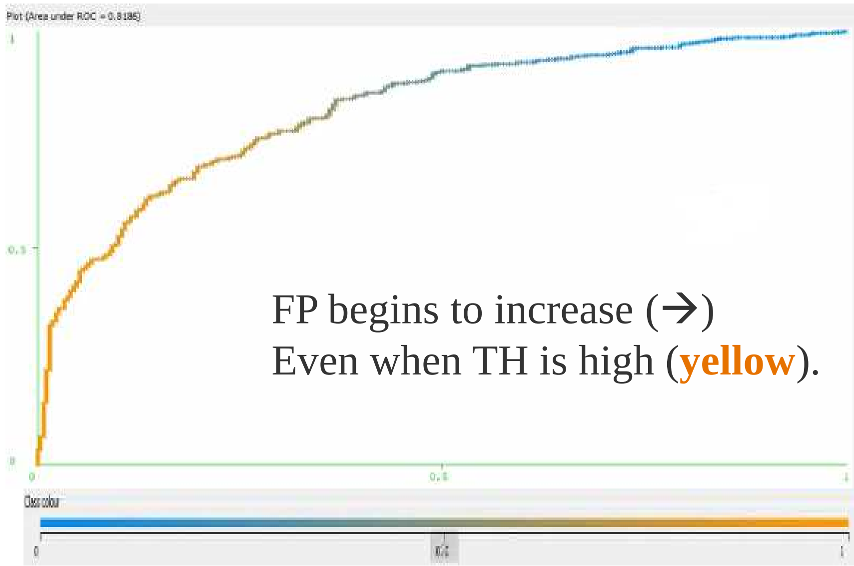
```
n_classes = 2
fpr = dict(); tpr = dict(); thresholds = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], thresholds[i] = roc_curve(y[i,:], scores[i, :] , pos_label=1)
    roc_auc[i] = auc(fpr[i], tpr[i])
    print(fpr[i], tpr[i], thresholds[i], auc(fpr[i], tpr[i]))
fpr, tpr, thresholds = metrics.roc_curve(y, scores, pos_label=2)
[ 0.   0.5  0.5  1. ] [ 0.5  0.5  1.   1. ] [ 0.9   0.65  0.6   0.2 ] 0.75
[ 0.   0.5  0.5  1. ] [ 0.5  0.5  1.   1. ] [ 0.8   0.4   0.35  0.1 ] 0.75
```

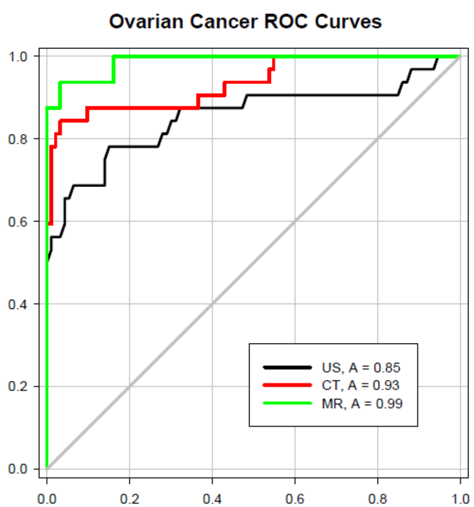
for binary class only can ignore this one

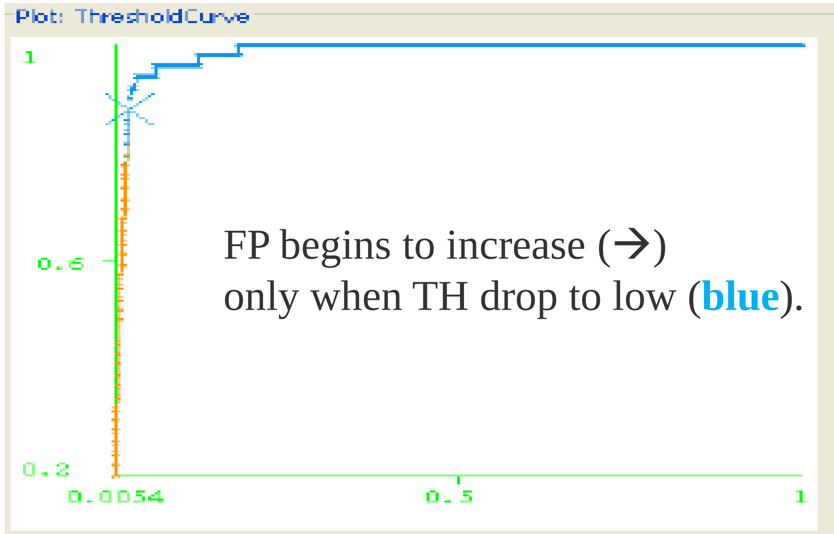


Which ROC Curves Are Better? Color ROC Curves

- Color **blue** = lower thresholds.







Cost vs. Accuracy[1]

Instead of considering only entropy or GINI!!

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS		+	–
	+	-1	100
	–	1	0

	PREDICTED CLASS		
ACTUAL CLASS		Yes	No
	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

- Example– note lower cost is better
  - M_2 has ↑ TP and ↓ FP, but it also has ↑ FN which is expensive

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	–
	+	150	**40**
	–	**60**	250

Accuracy = (150+250) / 500 = 80%

Cost = –150 + 60 +4000 = 3910

Model M_2	PREDICTED CLASS		
ACTUAL CLASS		+	–
	+	250	**45**
	–	**5**	200

Accuracy = (250+200) / 500 = 90%

Cost = –250 + 5 +4500 = 4255

Better accuracy, worse cost!!

Cost vs. Accuracy[2]

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS		+	–
	+	-1	100
	–	1	0

	PREDICTED CLASS		
ACTUAL CLASS		Yes	No
	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

- Example– note lower cost is better
  - M_2 has ↑ TP and ↓ FP, but it also has ↑ FN which is expensive

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	–
	+	150	**40**
	–	**60**	250

Accuracy = (150+250) / 500 = 80%

Cost = –150 + 60 +4000 = 3910

Model M_2	PREDICTED CLASS		
ACTUAL CLASS		+	–
	+	180	**10**
	–	**200**	110

Accuracy = (180+110) / 500 = 58%

Cost = –180 + 200 + 1000 = **1020**

Worse accuracy, better cost!!

Missing Values

- Simply remove the samples that have missing values?

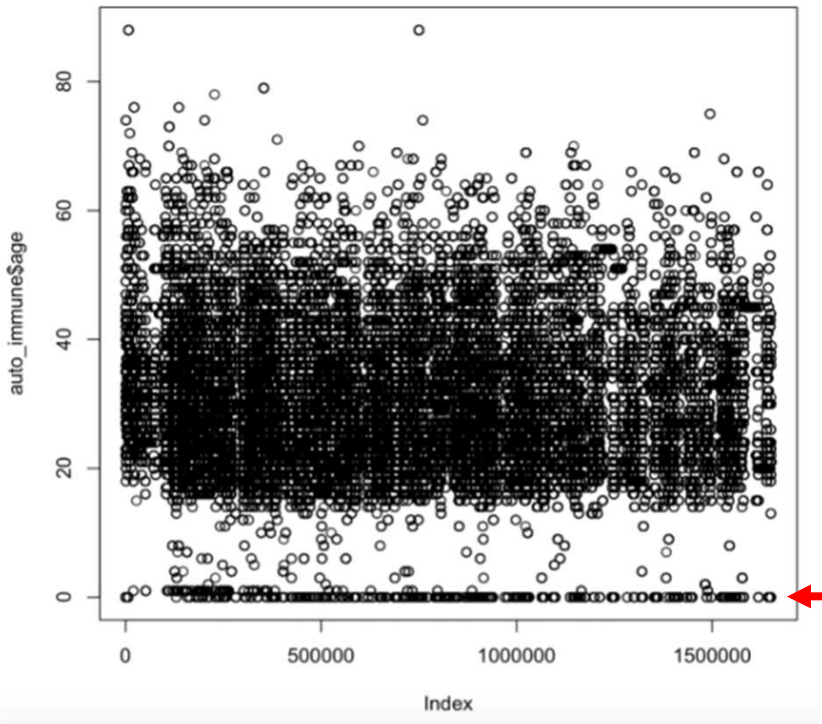
- Why do we have missing values?
  - Some measurements are not possible, e.g. plants died early
  - Equipment changes in experimental design, collecting different data
  - Refuse to answer, e.g. income, age, …
    - Bank has many customer records with missing ages… just remove those records ?

- Missing value may have significant implications, so …
  1) Assign the most common value of the attribute
     - Few plants die early, so assign most common value to an attribute

  2) Code "missing value" as an additional value (e.g. unknown, irrelevant)
     - "missing values" actually indicates some decisions are taken
     - e. g. missing medical test due to "too much pain" or "too expensive"…

Another Form of Missing Data

- NOT "***physically***" missing in the dataset.
- So, visualization may help.



many AGE at 0

Unraveling the autoimmune mystery, DM-01-18S

Rudra Panda, Marcie Tietjen, Indu Tiwari,
Jeff Rodabaugh, Ken Tamura

Compare Classification using CFM & F-Measure??

- Same accuracy & F-measures from 2 models.

Predicted Classes (BTs + Test Data + 25 Vars)

	NB	EW	NA	RM	BL
NB	6	0	0	0	0
EW	0	5	0	1	0
NA	0	1	0	3	1
RM	0	0	0	5	0
BL	0	0	0	0	3

24% off diagonal

BT on test data, all vars 2308

	NB	EW	NA	RM	BL
NB	5	0	0	1	0
EW	0	6	0	0	0
NA	0	2	0	3	0
RM	0	0	0	5	0
BL	0	0	0	0	3

24% off diagonal

- Which model is really better when 24% error in both confusion matrices?
  - Model 1…
  - Model 2…

- So far… all the quality evaluation methods measure quality at the final snapshot.
- They do not consider the prediction quality of ****EACH**** instance.

Parallel Coordinate Plot

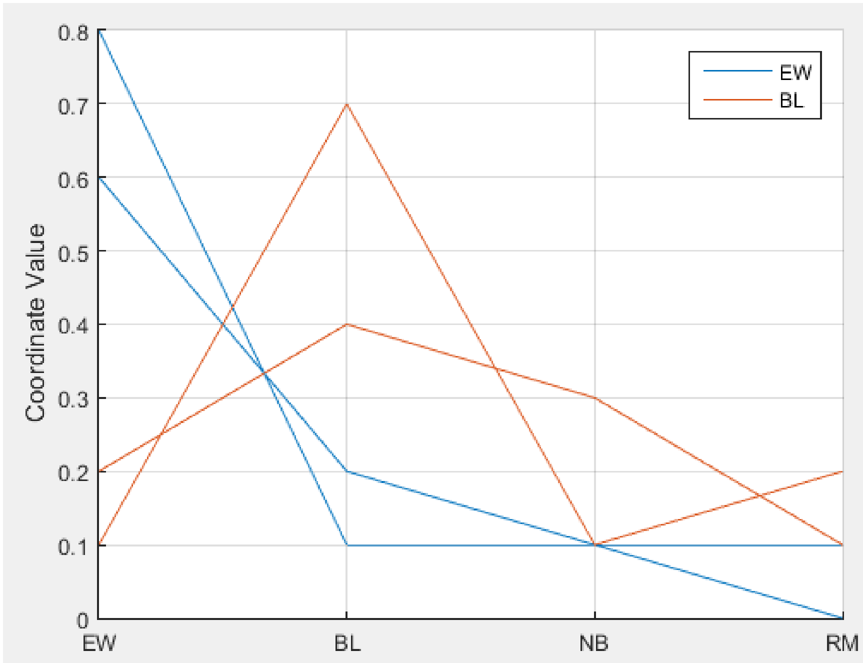
- Each **line** = one sample w/ **color** showing the **TRUE** class of the sample.
- The peak of each line shows the **predicted class** of the sample.

true labels / predicted score	EW	BL	NB	RM
Record 1 EW	0.8000	0.1000	0.1000	0
Record 2 EW	0.6000	0.2000	0.1000	0.1000
Record 3 BL	0.1000	0.7000	0.1000	0.2000
Record 4 BL	0.2000	0.4000	0.3000	0.1000

On Test Data
Predicted Classes, single DT

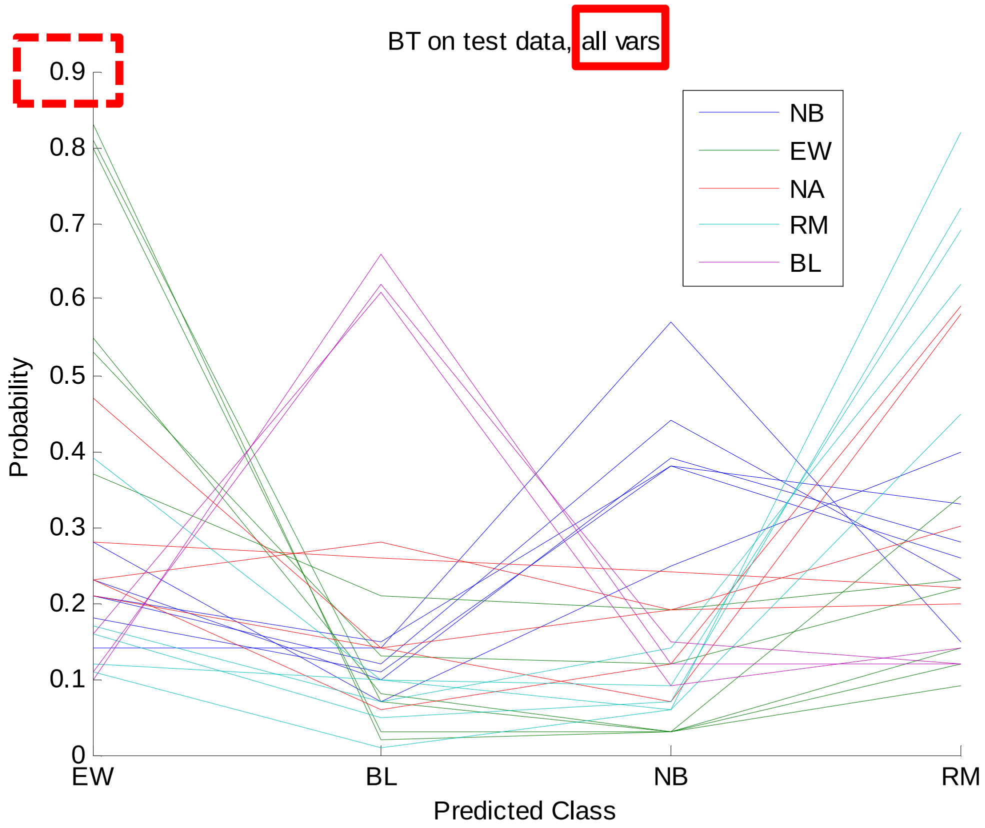
	NB	EW	NA	RM	BL
NB	5	0	0	1	0
EW	0	4	0	1	1
NA	1	0	0	4	0
RM	1	1	0	3	0
BL	0	0	0	0	3

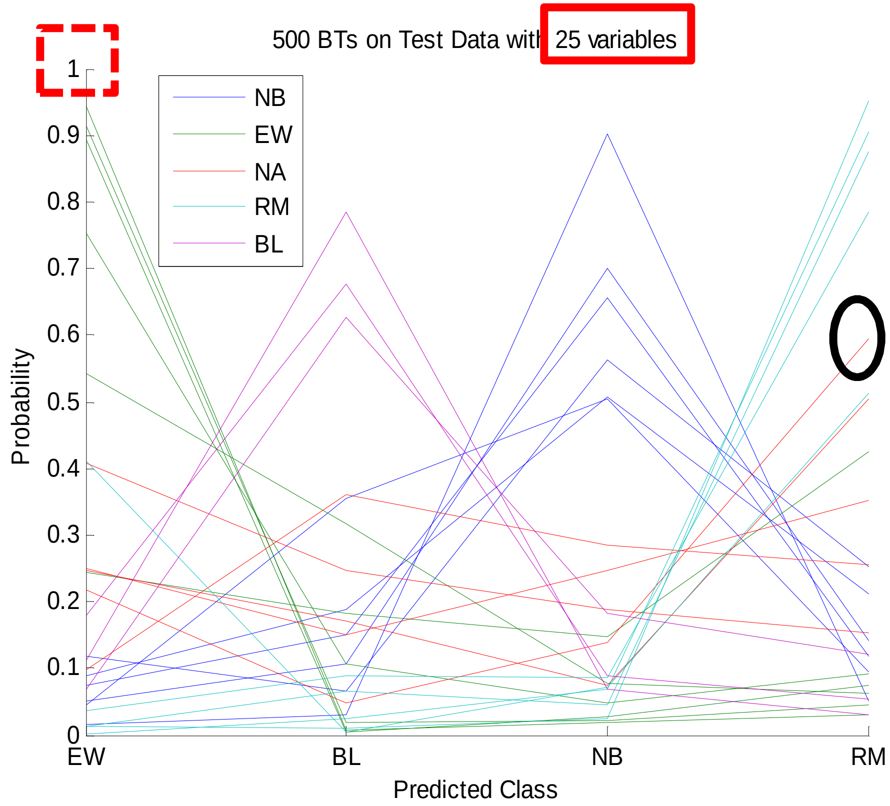
40% off diagonal



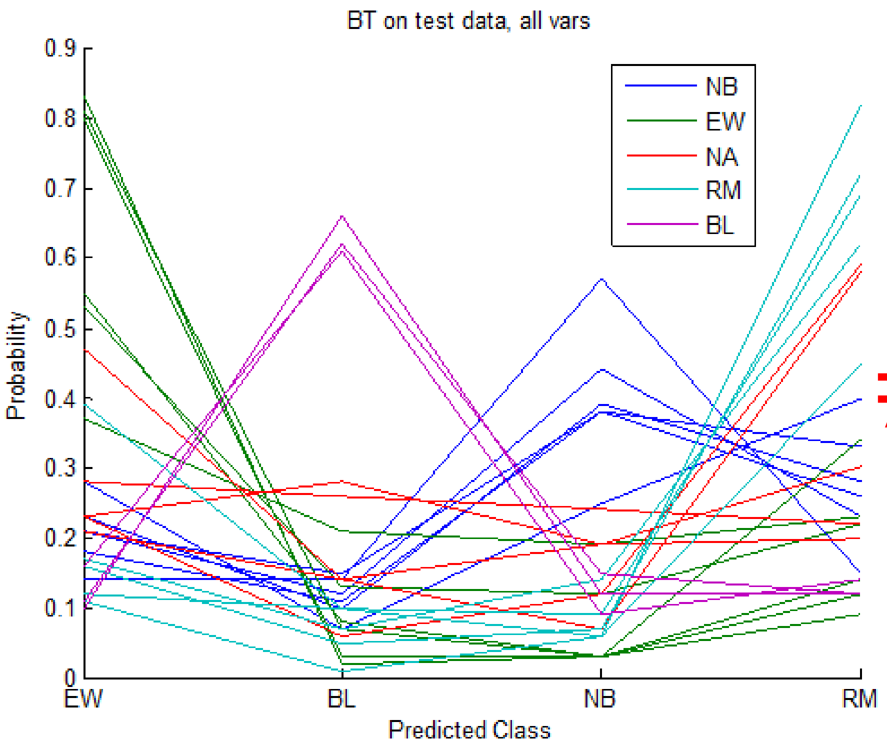
All Variables vs 25 Variables– PCP Comparison

- Which model is really better when 24% error in both confusion matrices?
- Confusion matrix does ****<u>NOT</u>**** tell us the whole story.
- Check PCP.

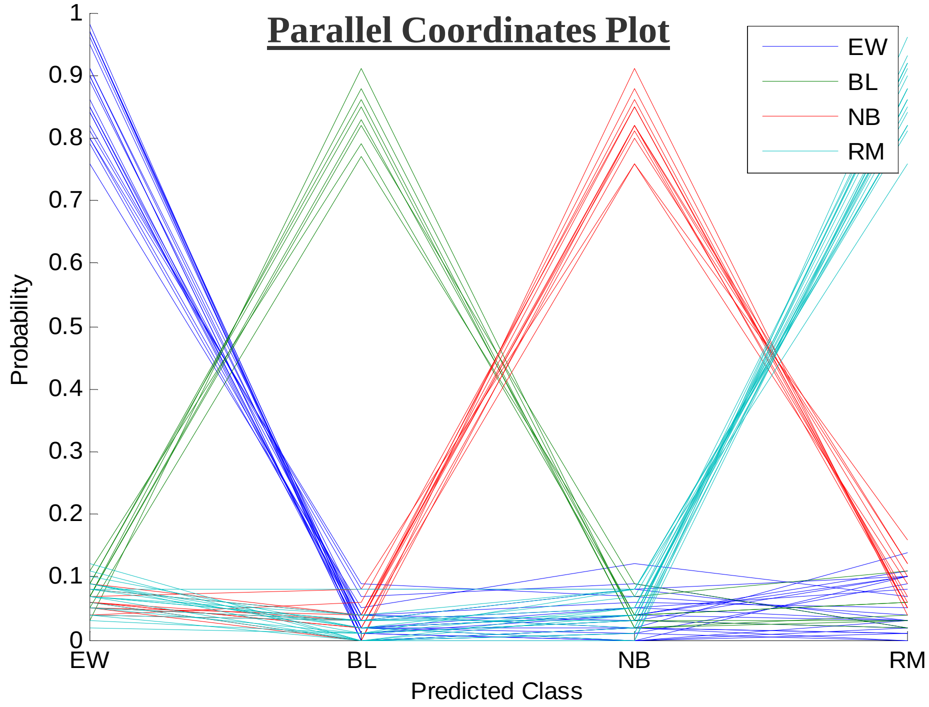




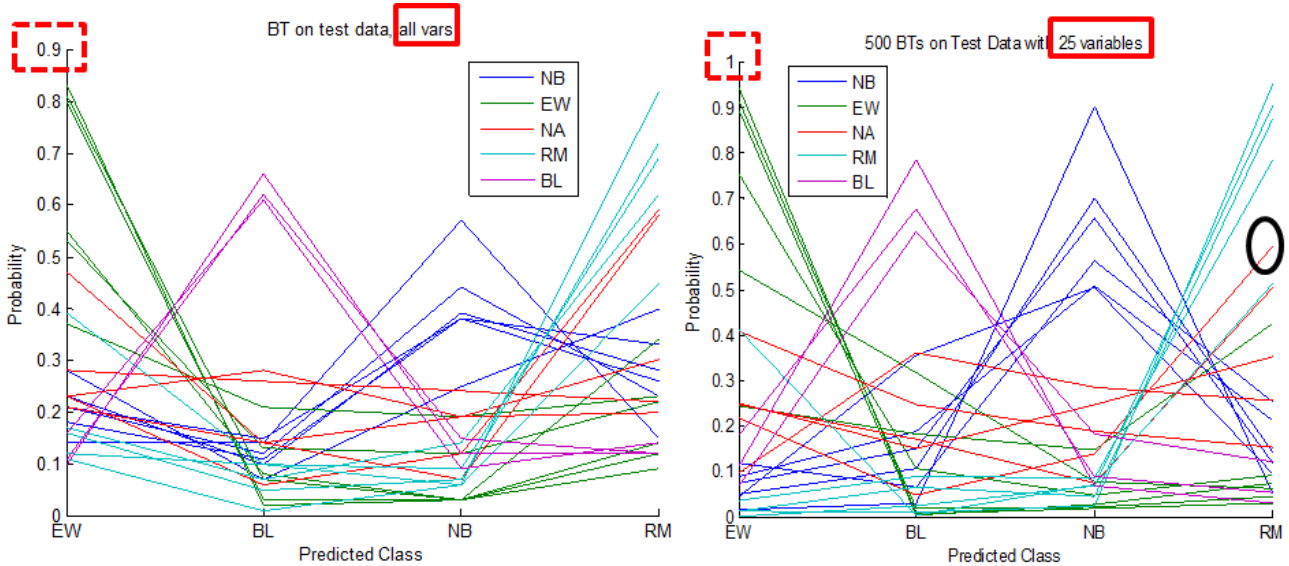
Another PCP Comparison



≠

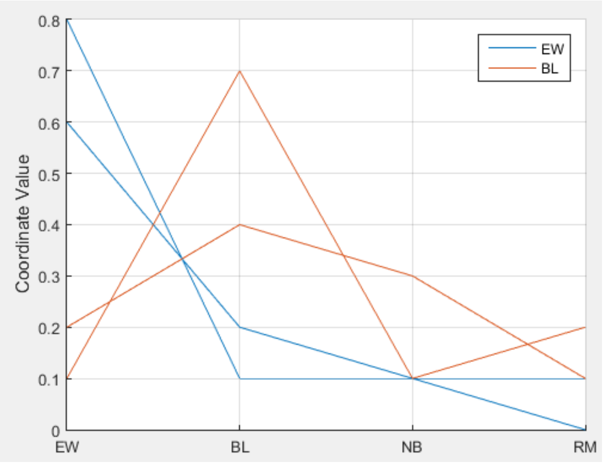


Which Model Has Better “PCPs”??



- $\text{Entropy}(p_1 \ldots p_n) = -\,p_1\log_2 p_1 - p_2\log_2 p_2 \ldots - p_n\log_2 p_n$

true labels / predicted score	EW	BL	NB	RM
Record 1 EW	0.8000	0.1000	0.1000	0
Record 2 EW	0.6000	0.2000	0.1000	0.1000
Record 3 BL	0.1000	0.7000	0.1000	0.2000
Record 4 BL	0.2000	0.4000	0.3000	0.1000



$$\text{Entropy}(p_1\ldots p_n) = -\,p_1\log_2 p_1 - p_2\log_2 p_2 \ldots - p_n\log_2 p_n$$

- 4 sample belong to **class a**, **b**, **c**, **d** ➔ **Lowest certainty** (or highest uncertainty)
  - Entropy(S) = $-0.25 \log_2 0.25 - 0.25 \log_2 0.25 - 0.25 \log_2 0.25 - 0.25 \log_2 0.25 = 2$

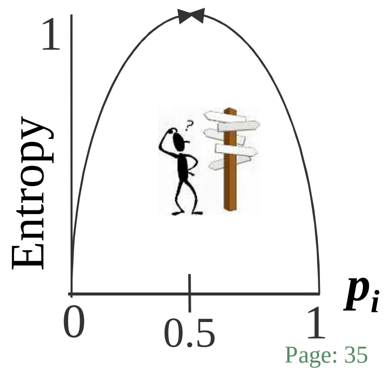
- 2 samples belong to **class a**, 2 samples belong **class b**
  - Entropy(S) = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 0.5 + 0.5 = 1$

- 3 samples belong to **class a**, 1 sample belongs **class b**
  - Entropy(S) = $-0.75 \log_2 0.75 - 0.25 \log_2 0.25 = 0.3 + 0.5 = 0.8$

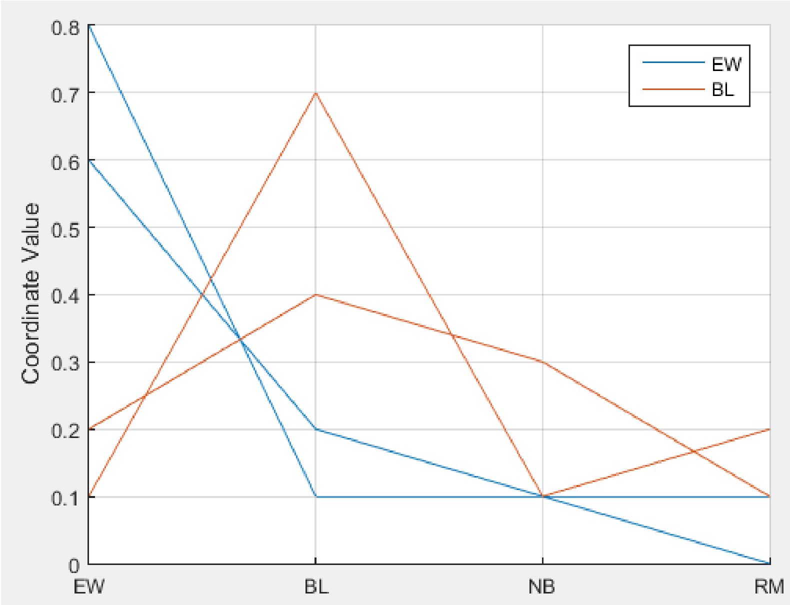
- ALL 4 samples belong to **class a**, NONE in **class b** ➔ **Highest certainty**
  - Entropy(S) = $-1 \log_2 1 - 0$ $\mathbf{\log_2 0}$ $= 0 + 0 = 0$
  - Need 0 bit to represent these 4 samples

- Smaller entropy ➔ more certain (pure) situation
- Larger entropy ➔ more uncertain (impure) situation

- Largest entropy for N classes is $\log_2 N$



Which Model Has Better "PCPs"?? Entropy Calculation



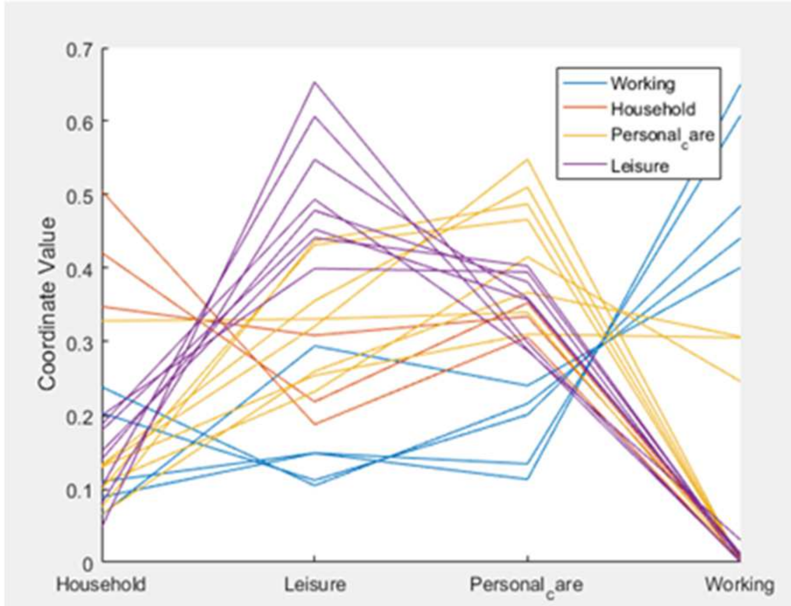
- $\text{Entropy}(p_1 \ldots p_n) = -\, p_1 \log_2 p_1 - p_2 \log_2 p_2 \ldots - p_n \log_2 p_n$

true labels / predicted score	EW	BL	NB	RM	
Record 1 EW	0.8000	0.1000	0.1000	0	$-0.8 \log_2 0.8 - 0.1 \log_2 0.1 - 0.1 \log_2 0.1 - 0 \log_2 0 = 0.9219$
Record 2 EW	0.6000	0.2000	0.1000	0.1000	$-0.6 \log_2 0.6 - 0.2 \log_2 0.2 - 0.1 \log_2 0.1 - 0.1 \log_2 0.1 = 1.571$
Record 3 BL	0.1000	0.7000	0.1000	0.2000	$-0.1 \log_2 0.1 - 0.7 \log_2 0.7 - 0.1 \log_2 0.1 - 0.2 \log_2 0.2 = 1.489$
Record 4 BL	0.2000	0.4000	0.3000	0.1000	$-0.2 \log_2 0.2 - 0.4 \log_2 0.4 - 0.3 \log_2 0.3 - 0.1 \log_2 0.1 = 1.8464$

Largest entropy for N = 4 classes is $\log_2 4 = 2$

Good Job?!

- Over 130,160 records.
- High accuracy 78%.
- Any comment?



Sinan Zhu, Qiong Yang, Uma Krishnaraju, Mohammad Khan, DM Project, 2016F

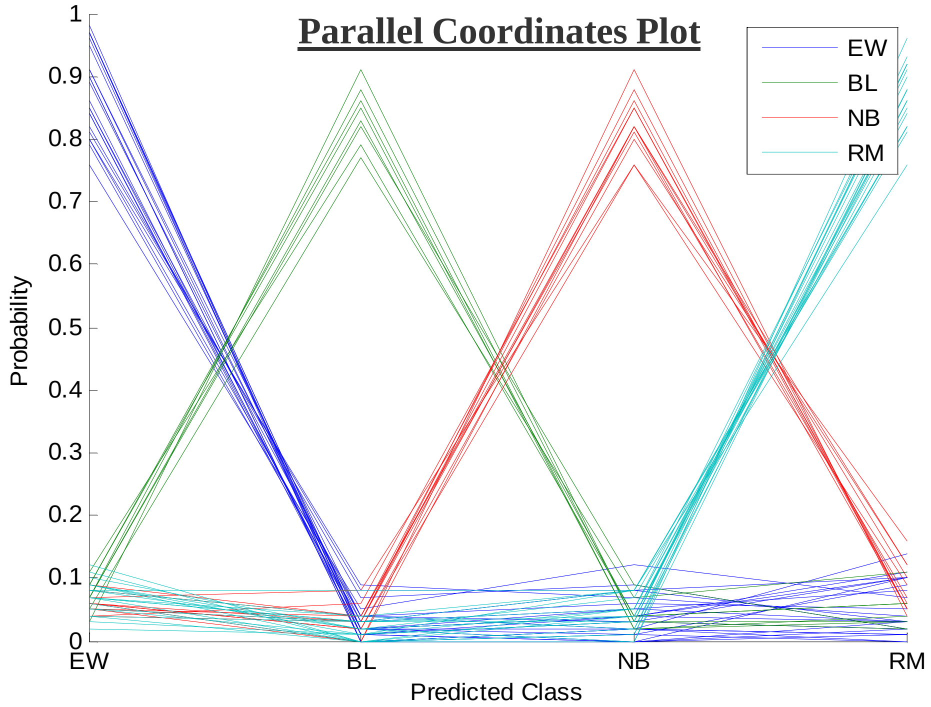
5499	12852	2586	10147
850	43776	21	292
2144	12172	3950	3988
4152	10193	1609	15919

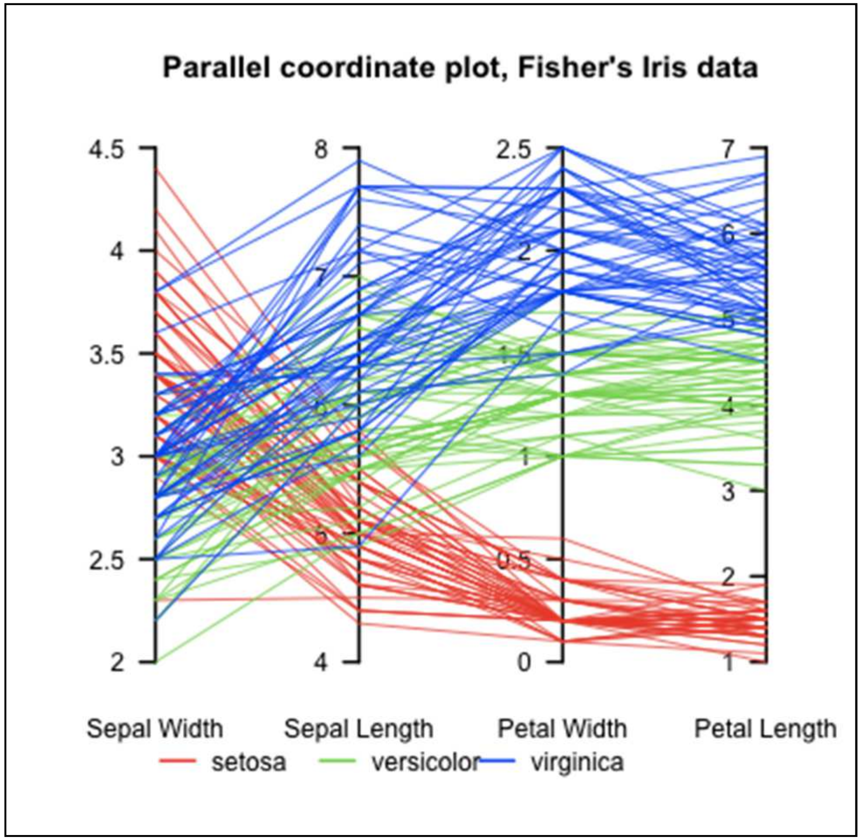
- Why only 24 lines over 130,160 records?
  - After discretizing numeric data into **VERY** few bins➔
  - Only 24 unique instances.
  - Much easier for machine to build high accuracy model.
  - Same as duplicating same data multiple times 🡐 **data cooking**.

Variation of Parallel Coordinate Plot

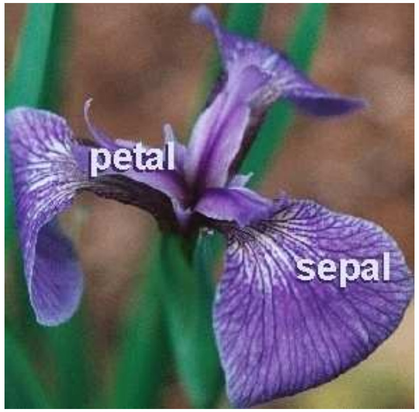
- Instead of line colors & ***x***-axis = target classes
- Set line colors = target classes & ***x***-axis = predictors to see ***class characteristics***.
  - Setosa tends to have smaller petal width and length, but large sepal width
  - Virginica tends to have larger petal width and length
  - We can also use group scatter plot (see later)

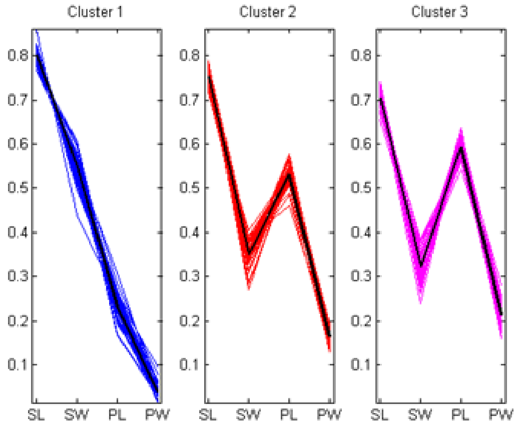
http://en.wikipedia.org/wiki/Parallel_coordinates





Sepal Length, Sepal Width, Petal Length”, Petal Width & Species



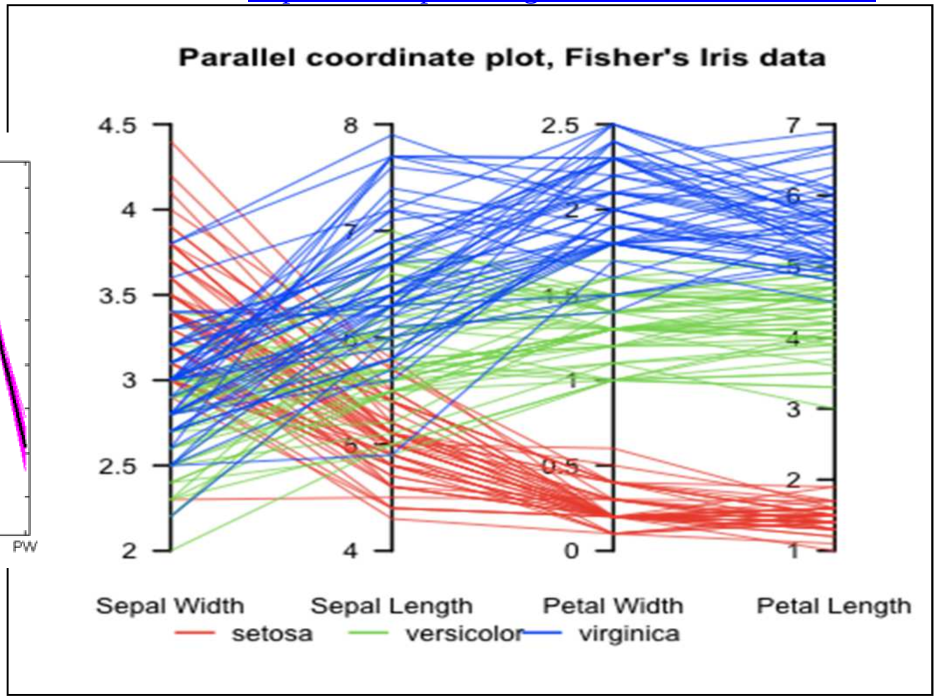


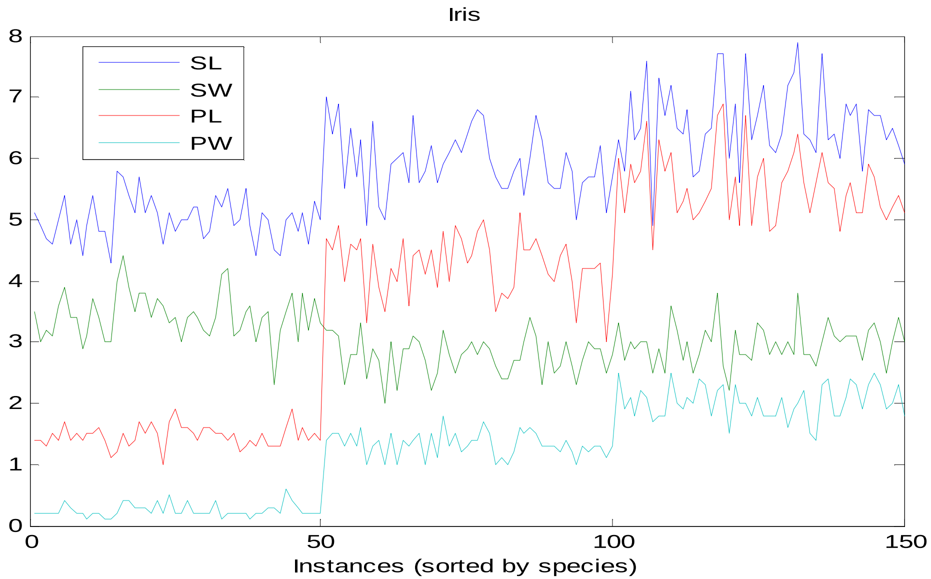
http://oranchak.com/?p=535

iris virginica iris versicolor iris setosa

	SL	SW	PL	PW	Species
1	5.1000	3.5000	1.4000	0.2000	versicolor
2	4.9000	3	1.4000	0.2000	setosa
3	4.7000	3.2000	1.3000	0.2000	versicolor
4	4.6000	3.1000	1.5000	0.2000	versicolor
5	5	3.6000	1.4000	0.2000	versicolor
6	5.4000	3.9000	1.7000	0.4000	versicolor
7	4.6000	3.4000	1.4000	0.3000	versicolor
8	5	3.4000	1.5000	0.2000	setosa
9	4.4000	2.9000	1.4000	0.2000	versicolor
10	4.9000	3.1000	1.5000	0.1000	setosa
11	5.4000	3.7000	1.5000	0.2000	versicolor
12	4.8000	3.4000	1.6000	0.2000	setosa

http://en.wikipedia.org/wiki/Parallel_coordinates





Setosa Versicolor Virginica

Matrix of Scatter Plots by Groups

- Setosa➔ petal width↓ & petal length↓, but sepal length↑
- Virginica➔ petal width↑ & petal length↑

http://oranchak.com/?p=535

iris virginica   iris versicolor   iris setosa