

DS 4400

Machine Learning and Data Mining I

Alina Oprea
Associate Professor
Khoury College of Computer Science
Northeastern University

October 15 2020

Outline

- Project discussion
- Evaluation of classifiers
 - Metrics
 - ROC curves
- Linear Discriminant Analysis (LDA)

Project Topic Discussion

- Room 1: Health
- Room 2: Image/Vision
- Room 3: Music
- Room 4: NLP
- Room 5: Sports/Finance

Accuracy and Error

Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\begin{aligned}\text{error} &= 1 - \frac{TP + TN}{P + N} \\ &= \frac{FP + FN}{P + N}\end{aligned}$$

Confusion Matrix

- Given a dataset of P positive instances and N negative instances:

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

$$\text{precision} = \frac{TP}{TP + FP}$$

Probability that classifier predicts positive correctly

$$\text{recall} = \frac{TP}{TP + FN}$$

Probability that actual class is predicted correctly

Classifiers can be tuned

- Logistic regression sets by default the threshold at 0.5 for classifying positive and negative instances
- Some applications have strict constraints on false positives (or other metrics)
 - Example: very low false positives in security (spam)
- Solution: choose different threshold

Probabilistic model $h_{\theta}(x) = P[y = 1|x; \theta]$

– Predict $y = 1$ if $h_{\theta}(x) \geq T$

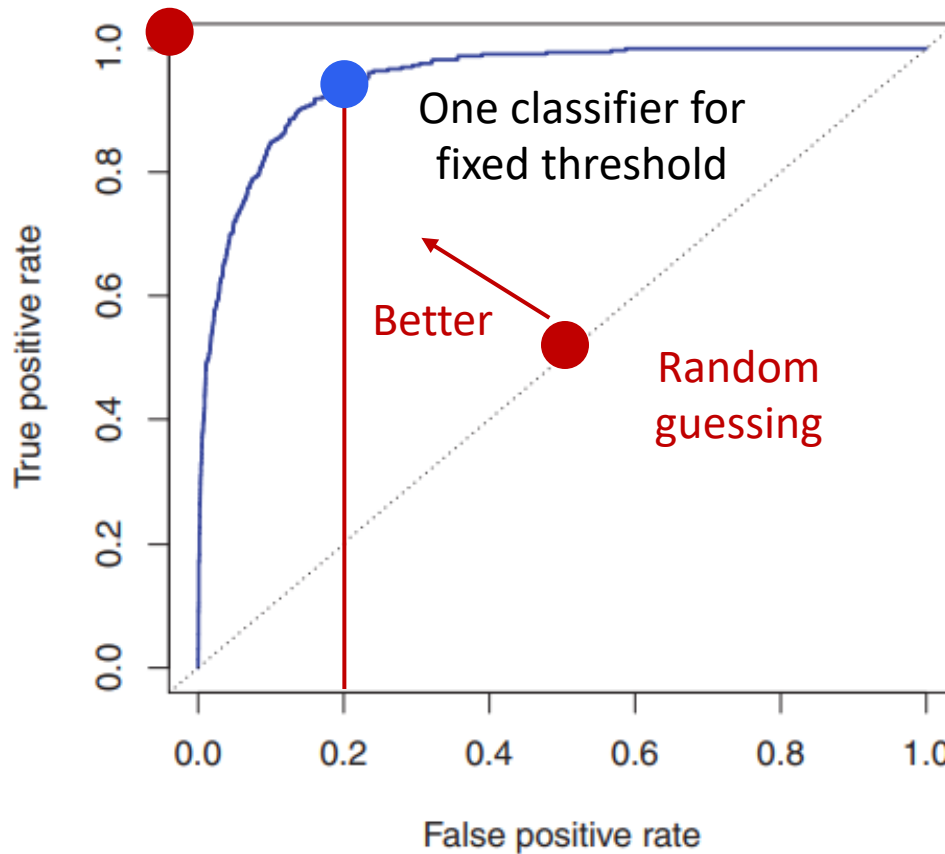
– Predict $y = 0$ if $h_{\theta}(x) < T$

Higher T , lower FP
Lower T , lower FN

ROC Curves

Perfect
classification

ROC Curve

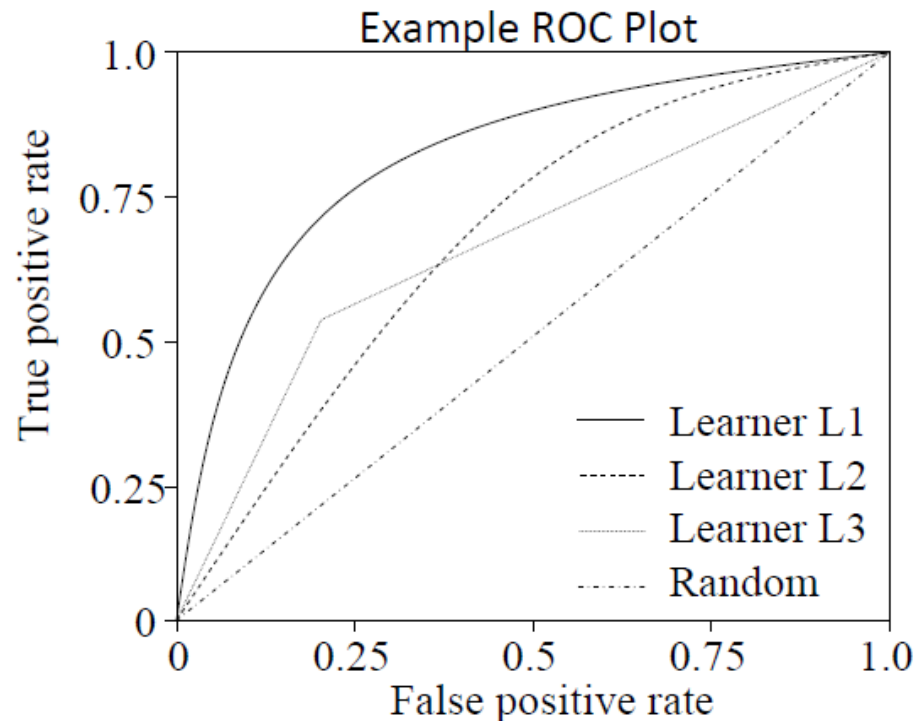


- Receiver Operating Characteristic (ROC)
- Determine operating point (e.g., by fixing false positive rate)

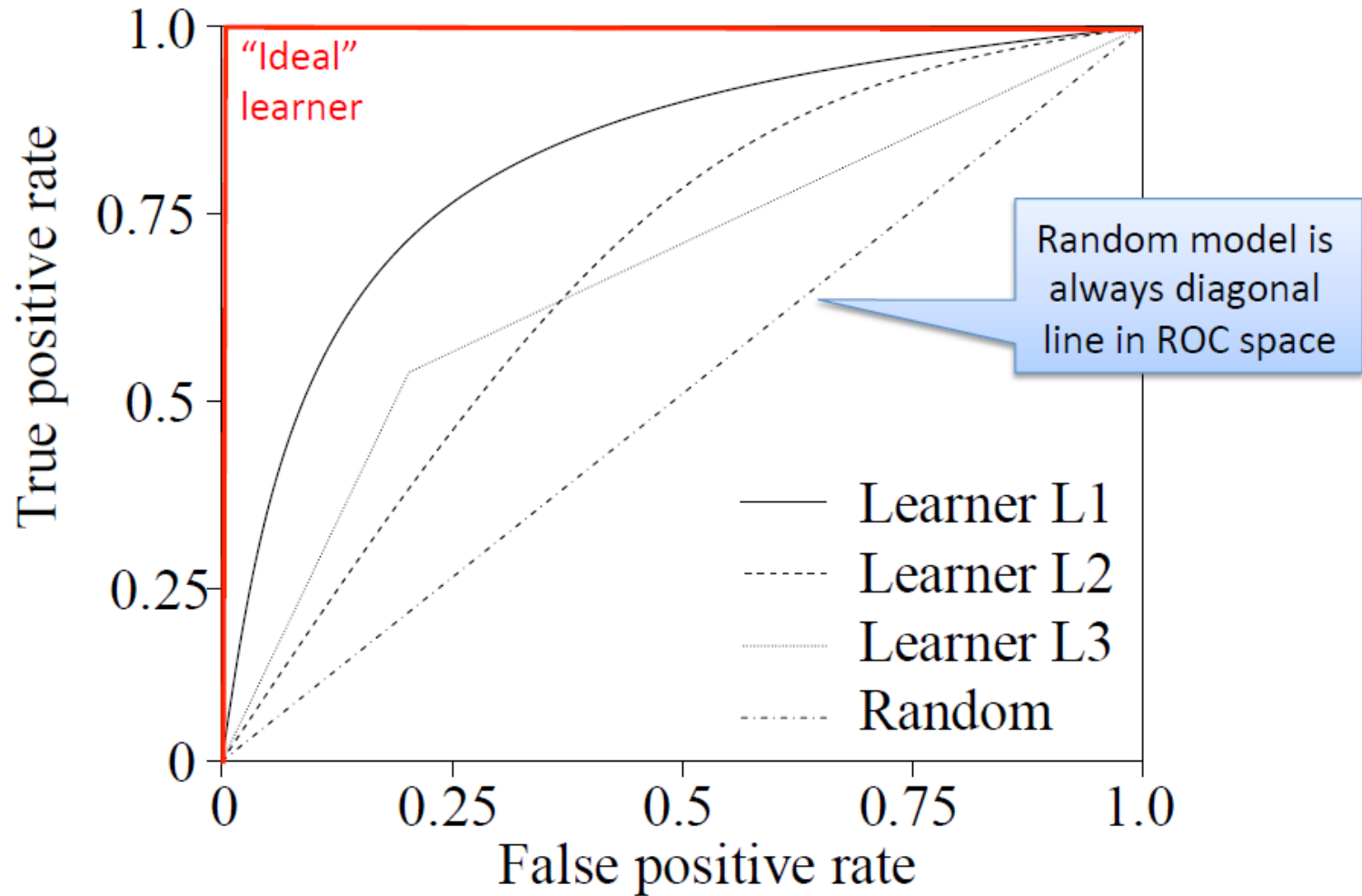
Performance Depends on Threshold

Predict positive if $P(y = 1 \mid \mathbf{x}) > \tau$ otherwise negative

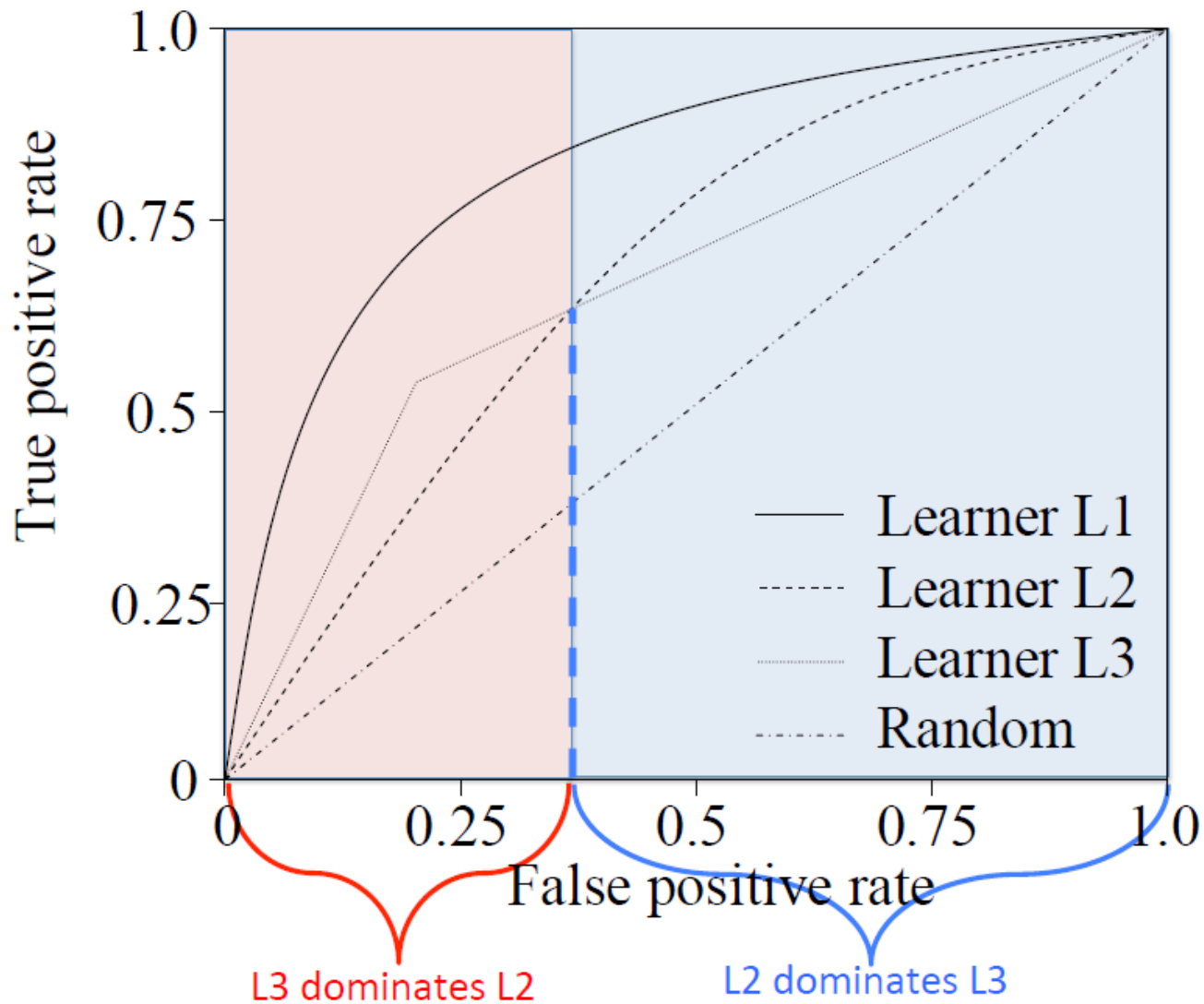
- Number of TPs and FPs depend on threshold τ
- As we vary τ we get different (TPR, FPR) points



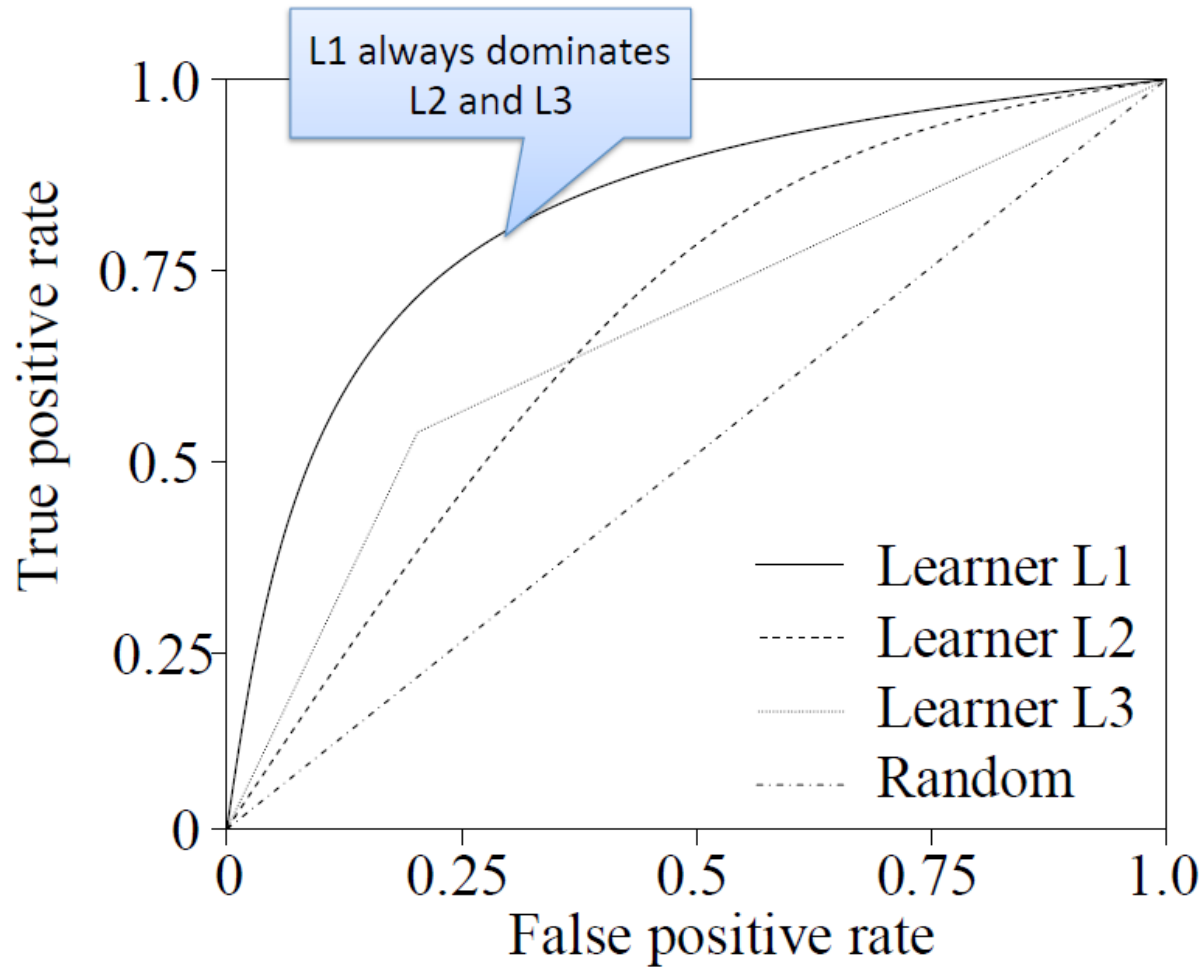
ROC Curve



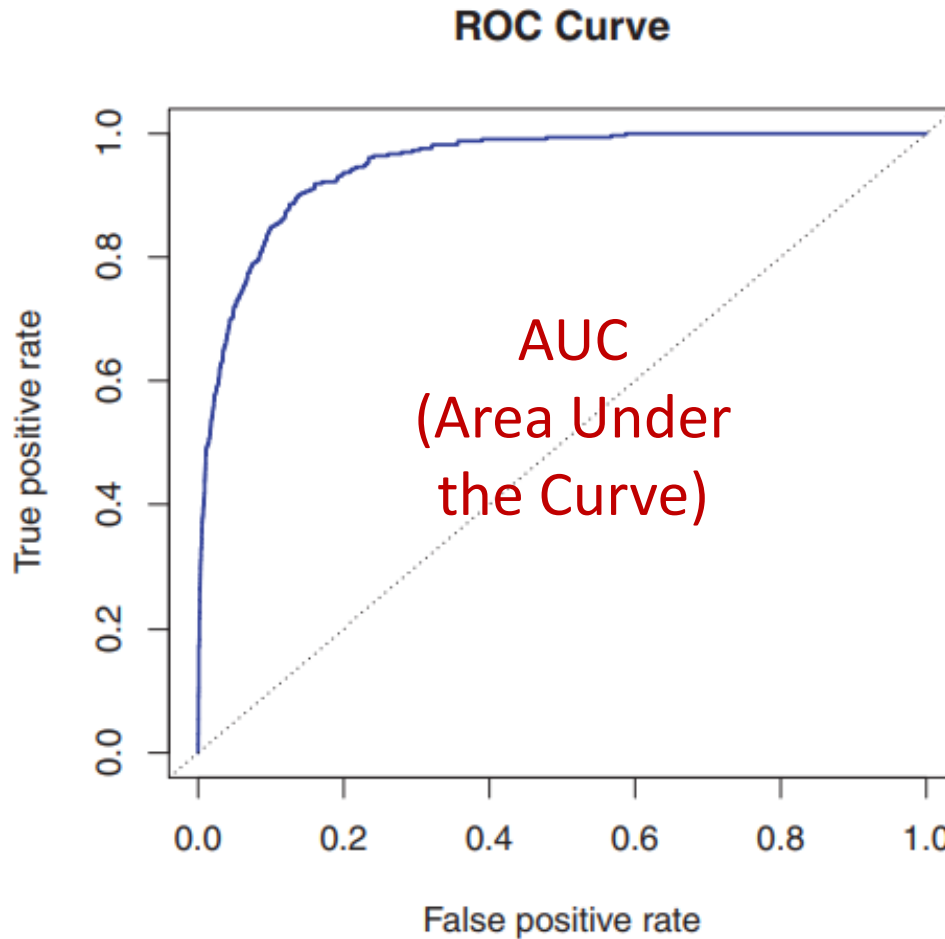
ROC Curve



ROC Curve



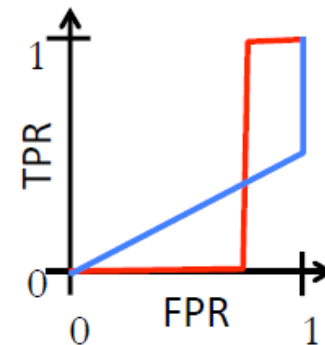
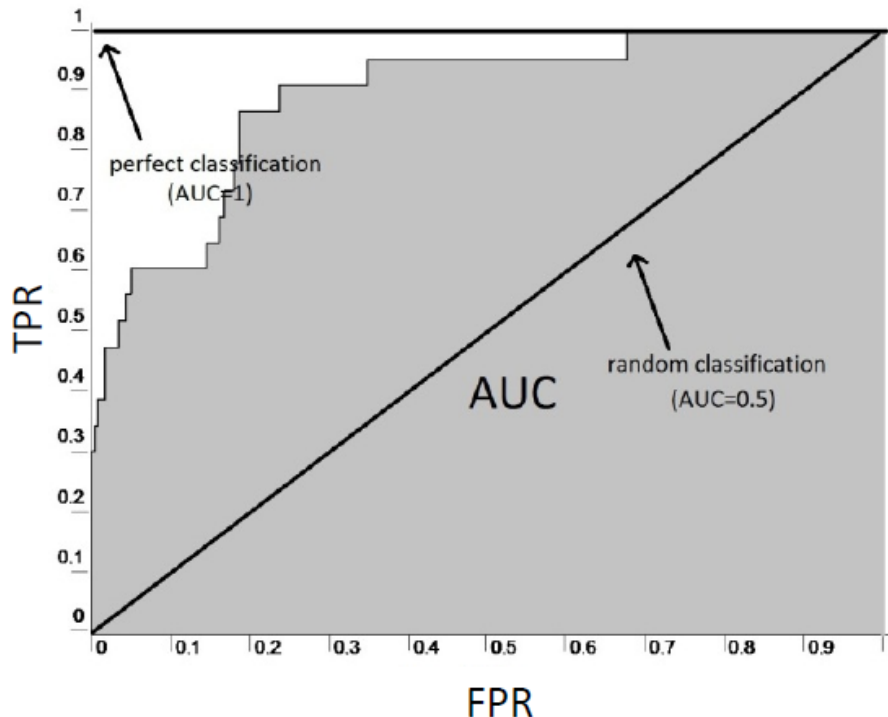
ROC Curves



- Another useful metric: Area Under the Curve (AUC)
- The closer to 1, the better!

Area Under the ROC Curve

- Can take area under the ROC curve to summarize performance as a single number
 - Be cautious when you see only AUC reported without a ROC curve; AUC can hide performance issues



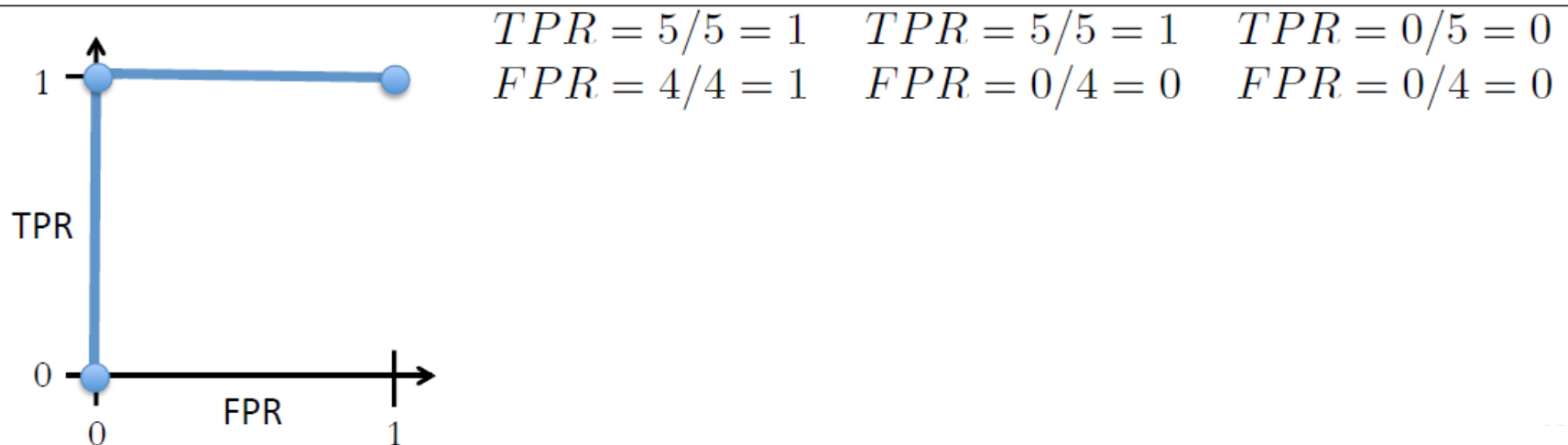
Same AUC, very different performance

ROC Curve Example

- Instructions
 - Use slides 15 and 16 from the PDF
 - Draw a ROC curve for each of these
 - There will be 3 points on each ROC curve, one for each threshold ($T = 0$, $T = 0.5$, $T = 1$)

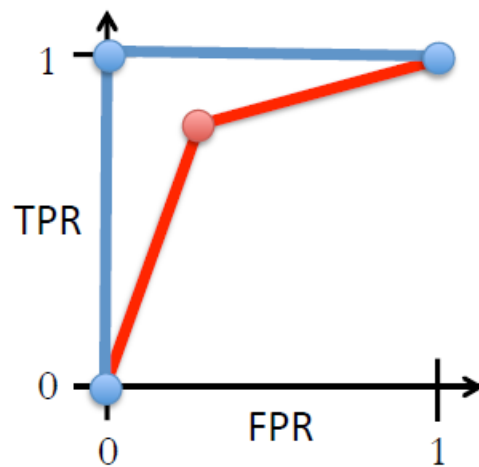
ROC Example

i	y_i	$p(y_i = 1 \mid \mathbf{x}_i)$	$h(\mathbf{x}_i \mid \theta = 0)$	$h(\mathbf{x}_i \mid \theta = 0.5)$	$h(\mathbf{x}_i \mid \theta = 1)$
1	1	0.9	1	1	0
2	1	0.8	1	1	0
3	1	0.7	1	1	0
4	1	0.6	1	1	0
5	1	0.5	1	1	0
6	0	0.4	1	0	0
7	0	0.3	1	0	0
8	0	0.2	1	0	0
9	0	0.1	1	0	0



ROC Example

i	y_i	$p(y_i = 1 \mid \mathbf{x}_i)$	$h(\mathbf{x}_i \mid \theta = 0)$	$h(\mathbf{x}_i \mid \theta = 0.5)$	$h(\mathbf{x}_i \mid \theta = 1)$
1	1	0.9	1	1	0
2	1	0.8	1	1	0
3	1	0.7	1	1	0
4	1	0.6	1	1	0
5	1	0.2	1	0	0
6	0	0.6	1	1	0
7	0	0.3	1	0	0
8	0	0.2	1	0	0
9	0	0.1	1	0	0



$$TPR = 5/5 = 1$$

$$FPR = 4/4 = 1$$

$$TPR = 4/5 = 0.8$$

$$FPR = 1/4 = 0.25$$

$$TPR = 0/5 = 0$$

$$FPR = 0/4 = 0$$

Linear Classifier Lab

```
: data = pd.read_csv('heart.csv')
data = data.dropna()
x_columns = data.columns != 'target'
data = utils.shuffle(data)
data.head()
```

:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	
	215	43	0	0	132	341	1	0	136	1	3.0	1	0	3	0
	145	70	1	1	156	245	0	0	143	0	0.0	2	0	2	1
	190	51	0	0	130	305	0	1	142	1	1.2	1	0	3	0
	90	48	1	2	124	255	1	1	175	0	0.0	2	2	2	1
	166	67	1	0	120	229	0	0	129	1	2.6	1	2	3	0

<https://www.kaggle.com/ronitf/heart-disease-uci>

Logistic Regression

```
: split = int(len(data) * 3/4)
x, y = data.loc[:, data.columns != 'target'], data['target']
x_train, x_test = x.iloc[:split], x.iloc[split:]
y_train, y_test = y.iloc[:split], y.iloc[split:]

logistic_model = LogisticRegression(max_iter=10000).fit(x_train, y_train)
print(len(data))
```

"""

```
pred_label = logistic_model.predict(x_test)
accuracy = logistic_model.score(x_test, y_test)
error = 1-accuracy
print("Accuracy=", accuracy)
print("Error=", error)
```

```
Accuracy= 0.8289473684210527
Error= 0.17105263157894735
```

Metrics

```
from sklearn.metrics import classification_report  
  
target_names = ['class 0', 'class 1']  
print(classification_report(y_test, pred_label, target_names=target_names))
```

	precision	recall	f1-score	support
class 0	0.86	0.79	0.83	39
class 1	0.80	0.86	0.83	37
accuracy			0.83	76
macro avg	0.83	0.83	0.83	76
weighted avg	0.83	0.83	0.83	76

ROC Curve

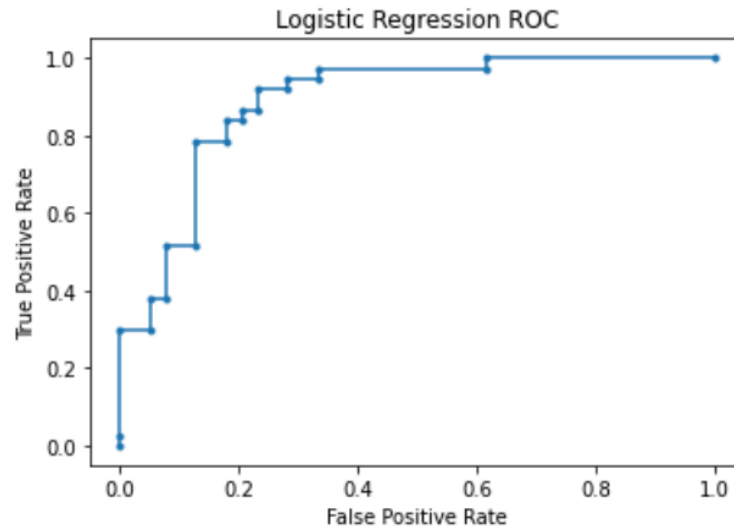
```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
from matplotlib import pyplot

pred_lr = logistic_model.predict_proba(x_test)
pred_lr = pred_lr[:, 1]
r_auc = roc_auc_score(y_test, pred_lr)
print("AUC=", r_auc)

lr_fpr, lr_tpr, _ = roc_curve(y_test, pred_lr)
pyplot.plot(lr_fpr, lr_tpr, marker='.', label='Logistic')
pyplot.xlabel('False Positive Rate')
pyplot.ylabel('True Positive Rate')
pyplot.title('Logistic Regression ROC')
```

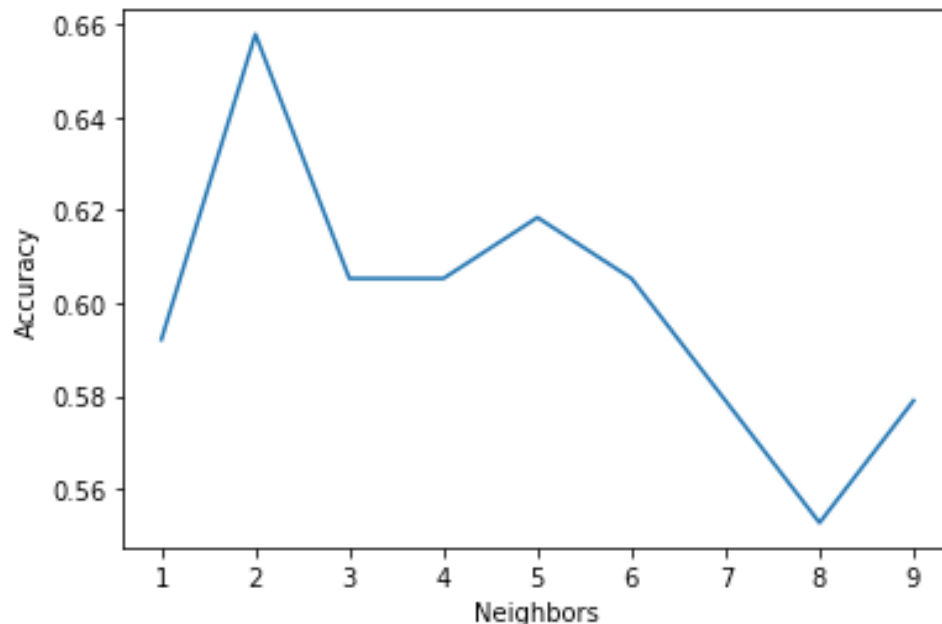
AUC= 0.8898128898128899

Text(0.5, 1.0, 'Logistic Regression ROC')



Lab kNN

```
from sklearn.neighbors import KNeighborsClassifier
accuracies = []
neighbors = list(range(1, 10))
knns = []
for n in neighbors:
    knn = KNeighborsClassifier(n_neighbors=n)
    knn.fit(x_train, y_train)
    knns.append(knn)
    accuracies.append(knn.score(x_test, y_test))
plt.figure().add_subplot(111, xlabel="Neighbors", ylabel="Accuracy")
plt.plot(neighbors, accuracies)
plt.show()
```



Acknowledgements

- Slides made using resources from:
 - Andrew Ng
 - Eric Eaton
 - David Sontag
- Thanks!