NBA 4920/6921 Lecture 6

Performance Metrics for Classification

Murat Unal

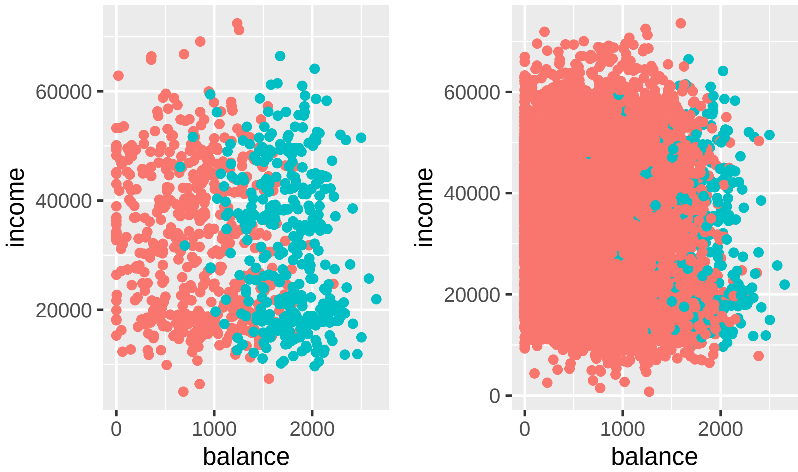
9/16/2021

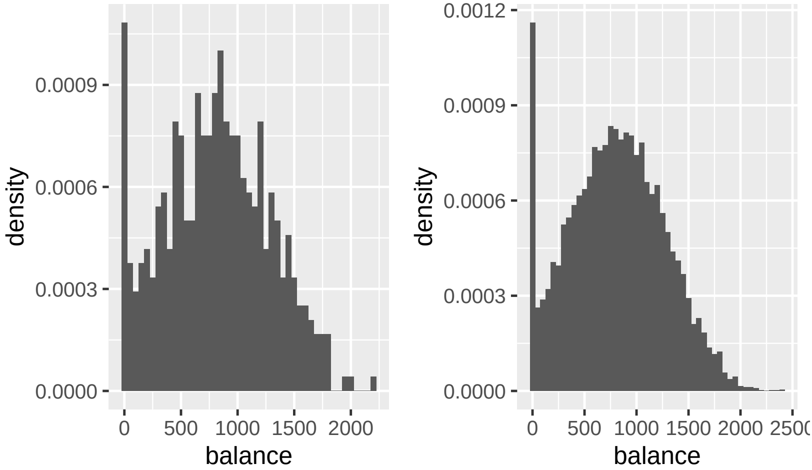
Agenda

- Quiz 5
- Reminders: Rmarkdown, Projects
- Demonstrating Sampling
- Confusion matrix
- Receiver Operating Characteristic (ROC) Curve

```r
rm(list=ls())
options(digits = 3, scipen = 999)
library(tidyverse)
library(ISLR)
library(cowplot)
library(ggcorrplot)
library(stargazer)
library(corrr)
library(lmtest)
library(sandwich)
library(MASS)
library(car)
library(jtools)
library(caret)
library(ROCR)
data <- ISLR::Default
auto <- ISLR::Auto
```

Sampling Works





density
0.0000
0.0003
0.0006
0.0009
0
500
1000
1500
2000
balance
density
0.0000
0.0003
0.0006
0.0009
0.0012
0
500
1000
1500
2000
balance

Confusion Matrix

Helps in diagnosing a model's prediction performance

		Actual Class	
		0	1
Predicted Class	0	N00	N10
	1	N01	N11

Number of observations $N = N00 + N10 + N01 + N11$

- True Negative: $N00$: Actual Class 0, Predicted Class 0
- False Positive: $N01$: Actual Class 0, Predicted Class 1
- False Negative: $N10$: Actual Class 1, Predicted Class 0
- True Positive: $N11$: Actual Class 1, Predicted Class 1

		Actual Class	
		0	1
Predicted Class	0	N00	N10
	1	N01	N11

- False Positive Rate: The fraction of negative examples that are classified as positive: $\frac{N01}{N00+N01}$
- False Negative Rate: The fraction of positive examples that are classified as negative: $\frac{N10}{N10+N11}$

		Actual Class	
		0	1
Predicted Class	0	N00	N10
	1	N01	N11

Accuracy: The share of correct predictions $= \frac{N00+N11}{N}$

Precision: The share of predicted positives ($\hat{Y} = 1$) that are correct. When the model predicts positive, how often is it correct?

$$P(Y = 1|\hat{Y} = 1) = \frac{N11}{N01+N11} = \frac{TP}{FP+TP}$$

Recall/Sensitivity/True Positive Rate: The share of positive outcomes ($Y = 1$) that we correctly predict

$$P(\hat{Y} = 1|Y = 1) = \frac{N11}{N10+N11} = \frac{TP}{FN+TP}$$

		Actual Class	
		0	1
Predicted Class	0	N00	N10
	1	N01	N11

Specificity: The share of neg. outcomes ($Y = 0$) that we correctly predict

$$P(\hat{Y} = 0 | Y = 0) = \frac{N00}{N01 + N00}$$

1 - **Specificity**= **False Positive Rate**

F1 Score: Seeks a balance between Precision and Recall.

A good F1 score means that you have low false positives and low false negatives.

An F1 score is considered perfect when it's 1, while the model is a total failure when it's 0.

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

So which criterion should we use? It depends.

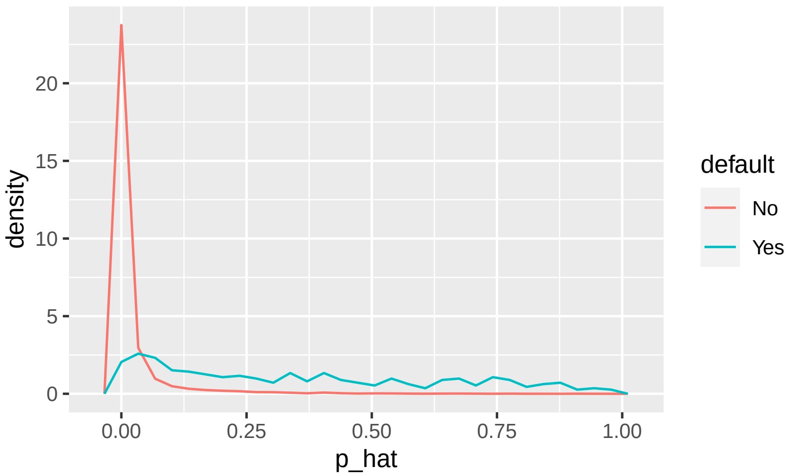
- **Accuracy**: if all errors are equal.
- **Precision**: if you want to have high confidence in predicted positives.
- **Recall/Sensitivity**: if true positives are more valuable than true negatives.
- **F1 Score**: if we need to seek a balance between Precision and Recall and there is an uneven class distribution (large number of actual negatives)

How did we do for predicting `default`?

```r
logit <- glm(default~balance+income+factor(student),
              family = "binomial",data =data)
# Predictions
p_hat <- predict(logit, type = "response")
# Add the predictions to the data:
data$p_hat <- p_hat
```

Visualize the density distributions for each class

```
ggplot(data,aes(y=..density..,x=p_hat,color=default))+
  geom_freqpoly()
```



```r
# Classify based on predictions
data$default <- ifelse(data$default=="Yes",1,0)
y_hat = as.numeric(p_hat >= 0.5)

# Create the confusion matrix
cm <- confusionMatrix(
  # Our predictions
  data = as.factor(y_hat),
  # Truth
  reference = as.factor(data$default),
  positive = "1"
)
cm$table
```

```
          Reference
Prediction    0    1
         0 9627  228
         1   40  105
```

```r
cm$table
```

```
          Reference
Prediction    0    1
         0 9627  228
         1   40  105
```

```r
cm$overall[c(1,5)]
```

```
    Accuracy AccuracyNull
       0.973        0.967
```

We predicted 97.32% of the observations correctly.

The default rate is only 3.33%. So, if we assigned "No" to everybody, we would have predicted 96.67% of the observations correctly.

```r
cm$table
```

```
          Reference
Prediction    0    1
         0 9627  228
         1   40  105
```

```r
cm$byClass[c(1,2,5,7)]
```

```
Sensitivity Specificity   Precision          F1
      0.315       0.996       0.724       0.439
```

What would happen if we changed the threshold from 0.5 to 0.25?

```r
# Classify based on predictions
y_hat2 = as.numeric(p_hat >= 0.25)

# Create the confusion matrix
cm2 <- confusionMatrix(
  # Our predictions
  data = as.factor(y_hat2),
  # Truth
  reference = as.factor(data$default),
  positive = "1"
)
cm2$table
```

```
          Reference
Prediction    0    1
         0 9477  149
         1  190  184
```

```r
cm$overall[c(1)]
```

```
Accuracy
   0.973
```

```r
cm2$overall[c(1)]
```

```
Accuracy
   0.966
```

```r
cm$byClass[c(1,2,5,7)]
```

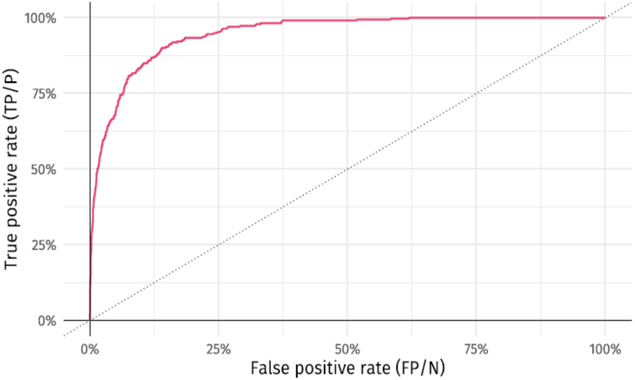
```
Sensitivity Specificity   Precision          F1
      0.315       0.996       0.724       0.439
```

```r
cm2$byClass[c(1,2,5,7)]
```

```
Sensitivity Specificity   Precision          F1
      0.553       0.980       0.492       0.521
```

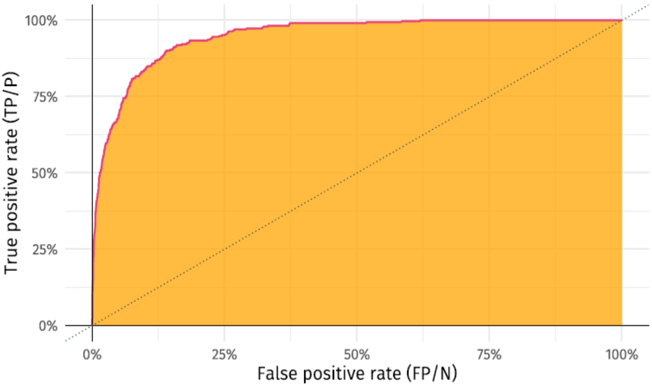
Receiver Operating Characteristic (ROC) Curve

ROC is a plot of signal (True Positive Rate) against noise (False Positive Rate).



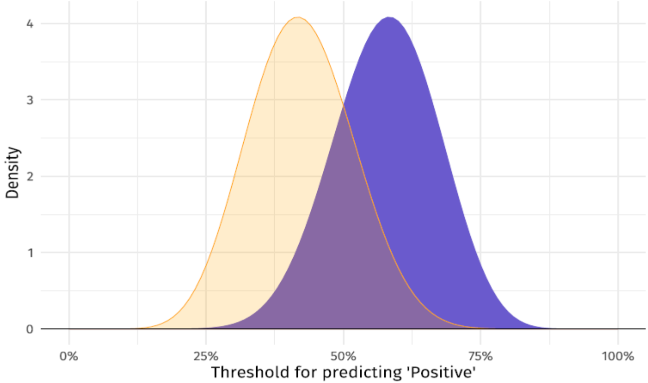
The model performance is determined by looking at the area under the ROC curve (or AUC).

The best possible AUC is 1 while the worst is 0.5 (the 45 degrees random line).

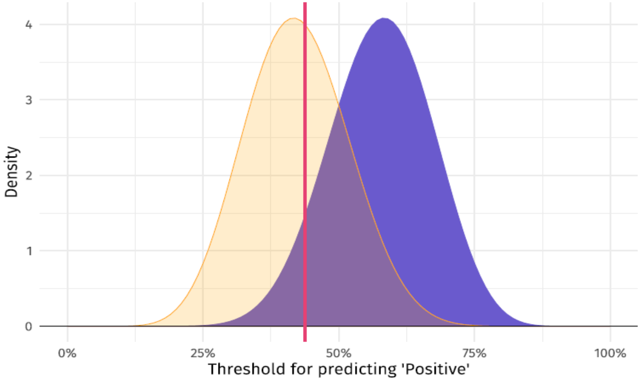


At its core AUC tells us how good we are in separating the two classes

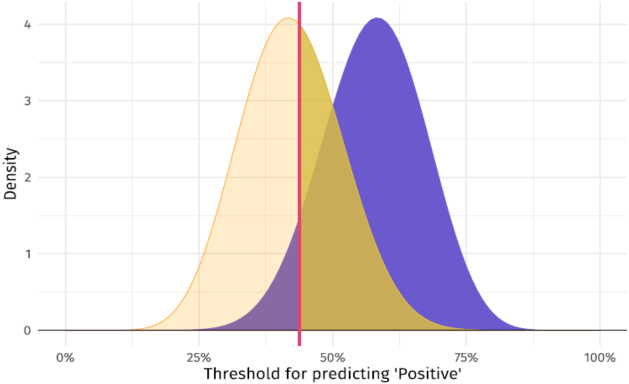
Example: Distribution of probabilities of two classes.



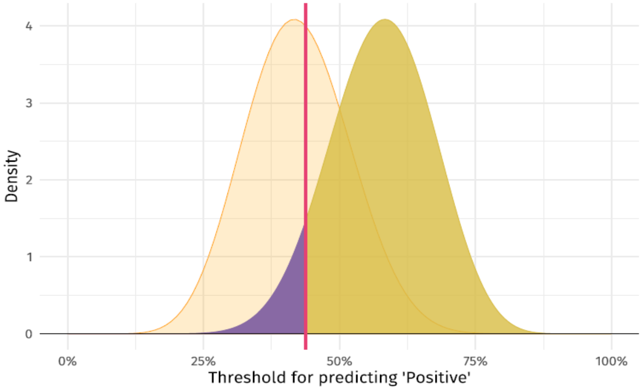
For any given threshold



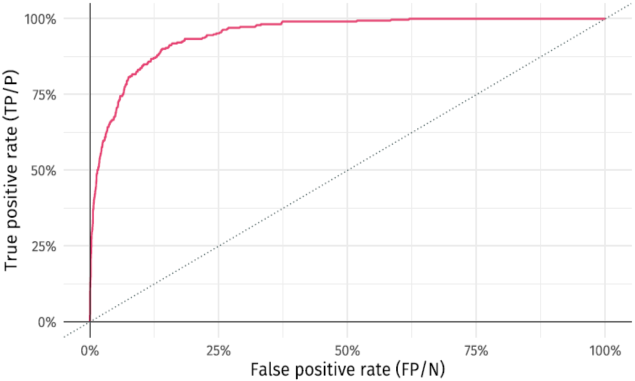
For any given threshold we get false positives



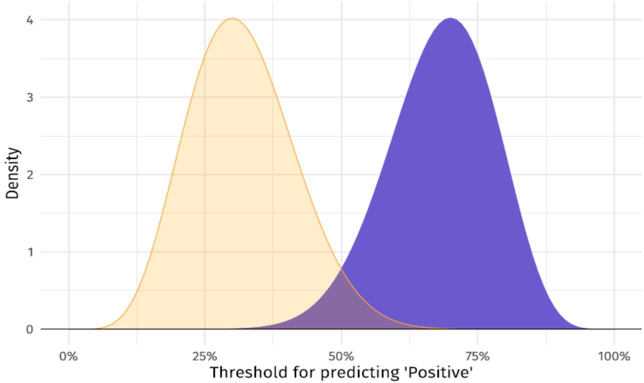
For any given threshold we get false positives and true positives



Moving through all possible thresholds generates the ROC curve

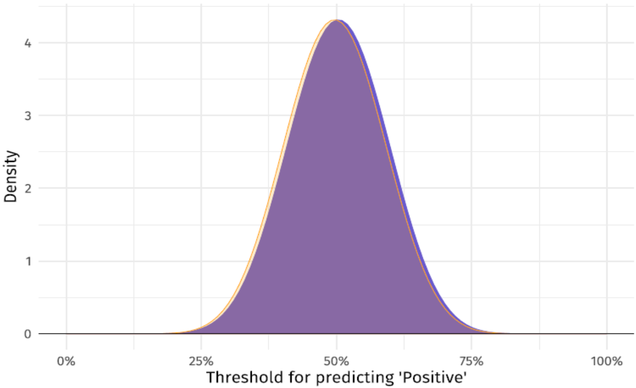


Increasing separation between positive and negative outcomes



Reduces error, shifts ROC, and increases AUC towards 1

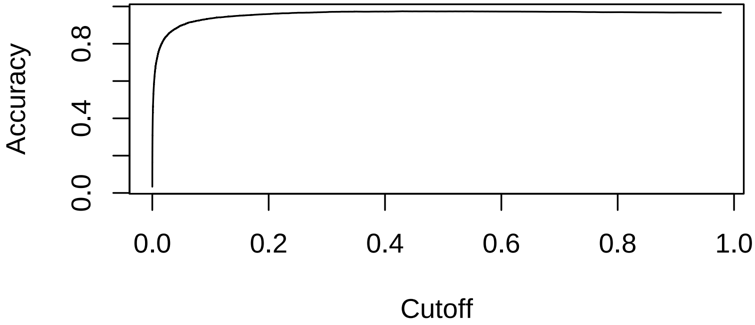
Failure in separation between positive and negative outcomes



Increases error, shifts ROC, and decreases AUC towards 0.5

Accuracy vs. cut-off values

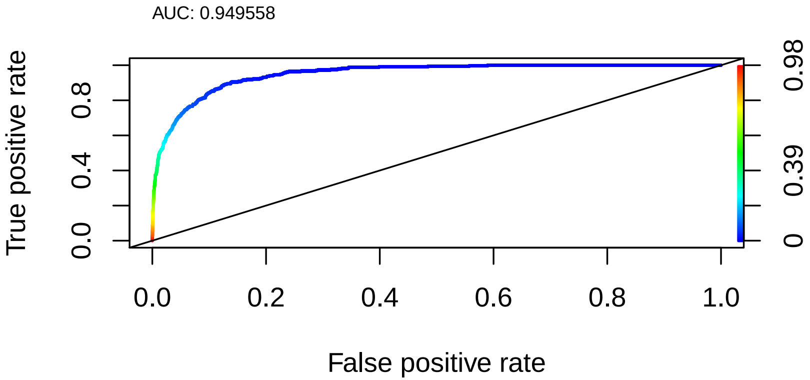
```
pred = prediction(p_hat, data$default)
perf = performance(pred, "acc")
plot(perf)
```



ROC curve and AUC

```
roc = performance(pred,"tpr","fpr")
plot(roc, colorize = T, lwd = 2)
abline(a = 0, b = 1)
auc = performance(pred, measure = "auc")
subtitle = sprintf("AUC: %f", auc@y.values)
mtext(side=3,line=1,at=0,adj=0,cex=0.7,subtitle)
```

ROC curve and AUC



Sources

1. The figures are from Ed Rubin's lecture notes.
   Ed Rubin (2020)
   Economics 524 (424): Prediction and Machine-Learning in Econometrics
   Univ, of Oregon

2. Notes are based on the book An Introduction to Statistical Learning (ISL)
   Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani (2017)
   https://www.statlearning.com/