# Performance Evaluation

David S. Rosenberg

Bloomberg ML EDU

October 26, 2017

# Baseline Models

# When is your prediction function good?

- Compare to previous models, if they exist.
- Is it good enough for business purposes?
- But also helpful to have some simple baseline models for comparison,
    - to make sure you're doing significantly better than trivial models
    - to make sure the problem you're working on even has a useful target function

# Zero-Information Prediction Function (Classification)

- For classification, let $y_{\text{mode}}$ be the most frequently occurring class in training.
- Prediction function that always predicts $y_{\text{mode}}$ is called
  - **zero-information prediction function**, or
  - **no-information prediction function**
- "No-information" because we're not using any information in input $x$.

# Zero-Information Prediction Function (Regression)

- What's the right zero-Information prediction function for **square loss**?
  - Mean of training data labels (See earlier homework.)
- What's the right zero-Information prediction function for **absolute loss**?
  - Median of training data labels (See earlier homework.)

# Single Feature Prediction Functions

- Choose a basic ML algorithm (e.g. linear regression or decision stumps)
- Build a set of prediction functions using ML algorithm, each **using only one feature**

# Regularized Linear Model

- Whatever fancy model you are using (gradient boosting, neural networks, etc.)
  - always spend some time building a linear baseline model
- Build a regularized linear model
  - lasso / ridge / elastic-net regression
  - $\ell_1$- and/or $\ell_2$- regularized logistic regression or SVM
- If your fancier model isn't beating linear,
  - perhaps something's wrong with your fancier model (e.g. hyperparameter settings), or
  - you don't have enough data to beat the simpler model
- Prefer simpler models if performance is the same
  - usually cheaper to train and easier to deploy

# Oracle Models

- Often helpful to get an upper bound on achievable performance.
- What's the best performance function you can get, looking at your validation data?
  - Performance will estimate the Bayes risk (i.e. optimal error rate).
  - This won't always be 0 - why?
- Using same model class as your ML model,
  - fit to the validation data without regularization.
  - Performance will tell us the limit of our model class, even with infinite training data.
  - Gives estimate of the approximation error of our hypothesis space.

# Describing Classifier Performance

# Confusion Matrix

- A **confusion matrix** summarizes results for a binary classification problem:

| | | Actual Class 0 | Actual Class 1 |
|---|---|---|---|
| Predicted | Class 0 | $a$ | $b$ |
| | Class 1 | $c$ | $d$ |

- $a$ is the number of examples of Class 0 that the classifier predicted [correctly] as Class 0.
- $b$ is the number of examples of Class 1 that the classifier predicted [incorrectly] as Class 0.
- $c$ is the number of examples of Class 0 that the classifier predicted [incorrectly] as Class 1.
- $d$ is the number of examples of Class 1 that the classifier predicted [correctly] as Class 1.

# Performance Statistics

- Many performance statistics are defined in terms of the confusion matrix.

| | | Actual | |
|---|---|---|---|
| | | Class 0 | Class 1 |
| Predicted | Class 0 | $a$ | $b$ |
| | Class 1 | $c$ | $d$ |

- **Accuracy** is the fraction of correct predictions:

$$\frac{a+d}{a+b+c+d}$$

- **Error rate** is the fraction of incorrect predictions:

$$\frac{b+c}{a+b+c+d}$$

# Performance Statistics

| | | Actual | |
|---|---|---|---|
| | | Class 0 | Class 1 |
| Predicted | Class 0 | $a$ | $b$ |
| | Class 1 | $c$ | $d$ |

- We can talk about accuracy of different subgroups of examples:
  - **Accuracy** for examples of **class** 0: $a/(a+c)$
  - **Accuracy** for examples **predicted to have class** 0: $a/(a+b)$.

# Issue with Accuracy and Error Rate

- Consider a **no-information classifier** that achieves the following:

| | | Actual | |
|---|---|---|---|
| | | Class 0 | Class 1 |
| Predicted | Class 0 | 10000 | 9 |
| | Class 1 | 0 | 0 |

- Accuracy is 99.9% and error rate is .09%.
- Two lessons:
  1. Accuracy numbers meaningless without knowing the **no-information rate** or **base rate**.
  2. Accuracy alone doesn't capture what's going on (0% success on class 1).

# Positive and Negative Classes

- So far, no class label has ever had any special status.
- In many contexts, it's very natural to identify a **positive class** and a **negative class**.
  - pregnancy test (**positive = you're pregnant)**
  - radar system (**positive = threat detected**)
  - searching for documents about bitcoin (**positive = document is about bitcoin**)
  - statistical hypothesis testing (**positive = reject the null hypothesis**)

# FP, FN, TP, TN

- Let's denote the **positive** class by $+$ and **negative** class by $-$:

| | | Actual | |
|---|---|---|---|
| | | Class $+$ | Class $-$ |
| Predicted | Class $+$ | TP | FP |
| | Class $-$ | FN | TN |

- TP is the number of **true positives**: predicted **correctly** as Class $+$.
- FP is the number of **false positives**: predicted **incorrectly** as Class $+$ (i.e true class $-$)
- TN is the number of **true negatives**: predicted **correctly** as Class $-$.
- FN is the number of **false negatives**: predicted **incorrectly** as Class $-$ (i.e. true class $+$)

# Precision and Recall

- Let's denote the **positive** class by $+$ and **negative** class by $-$:

| | | Actual | |
|---|---|---|---|
| | | Class $+$ | Class $-$ |
| Predicted | Class $+$ | TP | FP |
| | Class $-$ | FN | TN |

- The **precision** is the accuracy of the positive predictions: TP / (TP + FP)
  - High precision means low "**false alarm rate**" (if you test positive, you're probably positive)
- The **recall** is the accuracy of the positive class: TP/(TP+FN)
  - High recall means you're not missing many positives

# Information Retrieval

- Consider a database of 100,000 documents.
- Query for bitcoin returns 200 documents
- 100 of them are actually about bitcoin.
- 50 documents about bitcoin were not returned.

| | | Actual | |
|---|---|---|---|
| | | Class + | Class − |
| Predicted | Class + | 100 | 100 |
| | Class − | 50 | 99,750 |

# Precision and Recall

- Results from bitcoin query:

| | | Actual Class + | Actual Class − |
|---|---|---|---|
| Predicted | Class + | 100 | 100 |
| Predicted | Class − | 50 | 99,750 |

- The **precision** is the accuracy of the + predictions: TP / (TP + FP) $= 100/200 = 50\%$.
  - 50% of the documents offered as relevant are actually relevant.
- The **recall** is the accuracy of the positive class: TP/(TP+FN) $= 100/(100+50) = 67\%$.
  - 67% of the relevant documents were found (or "recalled").
- What's an easy way to get 100% recall?

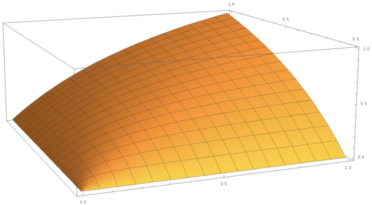
# $F_1$ score

- We really want high precision **and** high recall.
- But to choose a "best" model, we need a single number performance summary
- The **F-measure** or $F_1$ score is the **harmonic mean of precision and recall**:

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

- Ranges from 0 to 1.

$F_1(\text{precision}, \text{recall})$



| | Precision | Recall | $F_1$ |
|---|---|---|---|
| 1 | 0.01 | 0.99 | 0.02 |
| 2 | 0.20 | 0.80 | 0.32 |
| 3 | 0.40 | 0.90 | 0.55 |
| 4 | 0.60 | 0.62 | 0.61 |
| 5 | 0.90 | 0.95 | 0.92 |

# $F_\beta$ score

- Sometimes you want to weigh precision or recall more highly.
- $F_\beta$ score for $\beta \geqslant 0$:

$$F_\beta = (1+\beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

| | Precision | Recall | $F_1$ | $F_{0.5}$ | $F_2$ |
|---|---|---|---|---|---|
| 1 | 0.01 | 0.99 | 0.02 | 0.01 | 0.05 |
| 2 | 0.20 | 0.80 | 0.32 | 0.24 | 0.50 |
| 3 | 0.40 | 0.90 | 0.55 | 0.45 | 0.72 |
| 4 | 0.60 | 0.62 | 0.61 | 0.60 | 0.62 |
| 5 | 0.90 | 0.95 | 0.92 | 0.91 | 0.94 |

# TPR, FNR, FPR, TNR

| | | Actual | |
|---|---|---|---|
| | | Class + | Class − |
| Predicted | Class + | TP | FP |
| | Class − | FN | TN |

- **True positive rate** is the accuracy on the positive class: TP / (FN + TP)
  - same as **recall**, also called **sensitivity**
- **False negative rate** is the error rate on the positive class: FN / (FN + TP) (“**miss rate**”)
- **False positive rate** is error rate on the negative class: FP / (FP + TN)
  - also called **fall-out** or **false alarm rate**
- **True negative rate** is accuracy on the negative class: TN / (FP + TN) (“**specificity**”)

# Medical Diagnostic Test: Sensitivity and Specificity

- **Sensitivity** is another name for TPR and recall
  - What fraction of people with disease do we identify as having disease?
  - How "sensitive" is our test to indicators of disease?
- **Specificity** is another name for TNR
  - What fraction of people without disease do we identify as being without disease?
  - High specificity means few false alarms
- In medical diagnosis, we want both sensitivity and specificity to be high.

# Statistical Hypothesis Testing

- In a statistical hypothesis test, there are two possible actions:
  - reject the null hypothesis (Predict $+$), or
  - don't reject the null hypothesis (Predict $-$).
- Two possible error types are called "Type 1" and "Type 2" error.

| | | Actual: Class $+$ | Actual: Class $-$ |
|---|---|---|---|
| Predicted | Class $+$ | | Type 1 Error |
| | Class $-$ | Type 2 Error | |

# Thresholding Classification Score Functions

# The Classification Problem

- Action space $\mathcal{A} = \mathbf{R}$ Output space $\mathcal{Y} = \{-1, 1\}$
- **Real-valued prediction function** $f : \mathcal{X} \to \mathbf{R}$, called the **score function.**
- Convention was

$$f(x) > 0 \implies \text{Predict } 1$$
$$f(x) < 0 \implies \text{Predict } -1$$

# Example: Scores, Predictions, and Labels

| ID | Score | Predicted Class | True Class |
|---|---|---|---|
| 1 | -4.80 | - | - |
| 2 | -4.43 | - | - |
| 3 | -2.09 | - | - |
| 4 | -1.30 | - | - |
| 5 | -0.53 | - | + |
| 6 | -0.30 | - | + |
| 7 | 0.49 | + | - |
| 8 | 0.98 | + | - |
| 9 | 2.25 | + | + |
| 10 | 3.37 | + | + |
| 11 | 4.03 | + | + |
| 12 | 4.90 | + | + |

- Performance measures:
  - Error Rate $= 4/12 \approx .33$
  - Precision $= 4/6 \approx .67$
  - Recall $= 4/6 \approx .67$
  - $F_1 = 4/6 \approx .67$
- Now predict + iff Score>2?
  - Error Rate $= 2/12 \approx .17$
  - Precision $= 4/4 = 1.0$
  - Recall $= 4/6 \approx .67$
  - $F_1 = 0.8$
- Now predict + iff Score>-1?
  - Error Rate $= 2/12 \approx .17$
  - Precision $= 6/8 = .75$
  - Recall $= 6/6 = 1.0$
  - $F_1 = 0.86$

# Thresholding the Score Function

- Generally, different thresholds on the score function lead to
  - different confusion matrices
  - different performance metrics
- One should choose the threshold that optimizes the business objective.
- Examples:
  - Maximize $F_1$ (or $F_{0.2}$ or $F_{2.0}$, etc.)
  - Maximize Precision, such that Recall $> 0.8$.
  - Maximize Sensitivity, such that Specificity $> 0.7$.

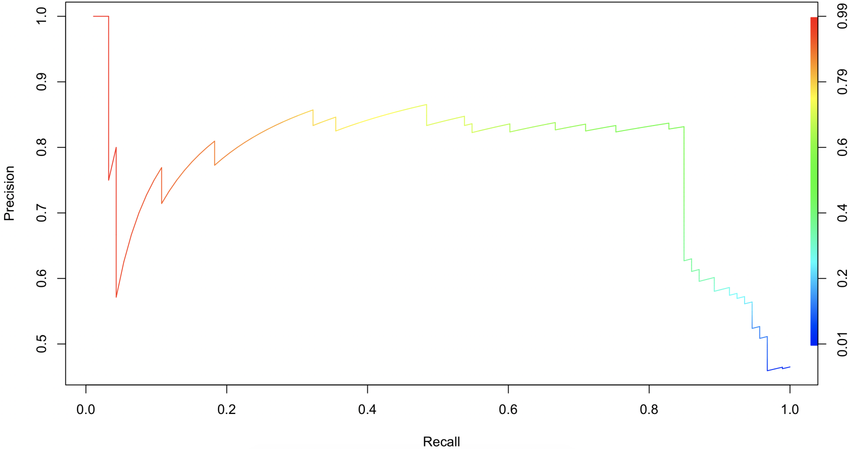
# The Performance Curves

# Precision-Recall as Function of Threshold

| ID | Score | Predicted Class | True Class |
|---|---|---|---|
| 1 | -4.80 | - | - |
| 2 | -4.43 | - | - |
| 3 | -2.09 | - | - |
| 4 | -1.30 | - | - |
| 5 | -0.53 | - | + |
| 6 | -0.30 | - | + |
| 7 | 0.49 | + | - |
| 8 | 0.98 | + | - |
| 9 | 2.25 | + | + |
| 10 | 3.37 | + | + |
| 11 | 4.03 | + | + |
| 12 | 4.90 | + | + |

- What happens to **recall** as we decrease threshold from $+\infty$ to $-\infty$?
  - Recall increases (or at least never decreases)
- What happens to **precision** as we decrease threshold from $+\infty$ to $-\infty$?
  - If score capture confidence,
    - we expect higher threshold to have higher precision.
  - But no guarantees in general.

# Precision-Recall Curve




- What threshold to choose? Depends on your preference between precision and recall.

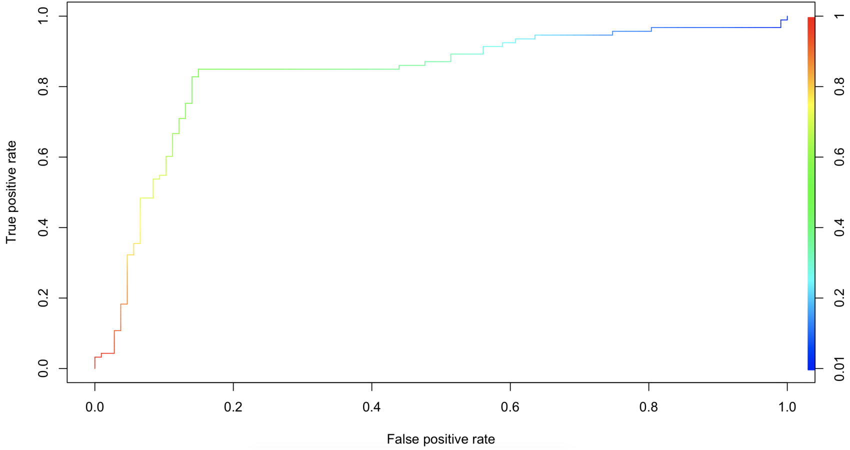
Example from ROCR Package.

# Receiver Operating Characteristic (ROC) Curve

| ID | Score | Predicted Class | True Class |
|---|---|---|---|
| 1 | -4.80 | - | - |
| 2 | -4.43 | - | - |
| 3 | -2.09 | - | - |
| 4 | -1.30 | - | - |
| 5 | -0.53 | - | + |
| 6 | -0.30 | - | + |
| 7 | 0.49 | + | - |
| 8 | 0.98 | + | - |
| 9 | 2.25 | + | + |
| 10 | 3.37 | + | + |
| 11 | 4.03 | + | + |
| 12 | 4.90 | + | + |

- Recall **FPR** and **TPR**:
  - FPR = FP / (Number of Negatives Examples)
  - TPR = TP / (Number of Positives Examples)
- As we decrease threshold from $+\infty$ to $-\infty$,
  - Number of positives predicted increases - some correct, some incorrect.
  - So both FP and TP increase.
- ROC Curve charts TPR vs FPR as we vary the threshold...

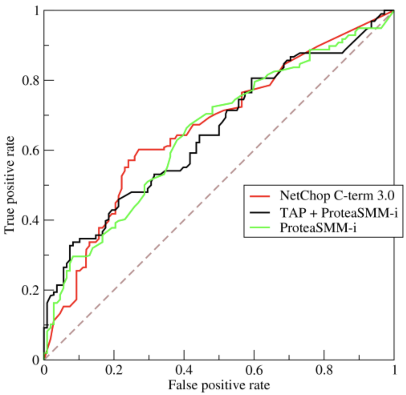
# Receiver Operating Characteristic (ROC) Curve




- Ideal ROC curve would go straight up on the left side of the chart.

Example from ROCR Package.

# Comparing ROC Curves




- Here we have ROC curves for 3 score functions.
- For different FPRs, different score functions give better TPRs.
- No score function dominates another at every FPR.
- Can we come up with an overall performance measure for a score function?

Figure by bOR from Wikimedia Commons.

# Area Under the ROC Curve

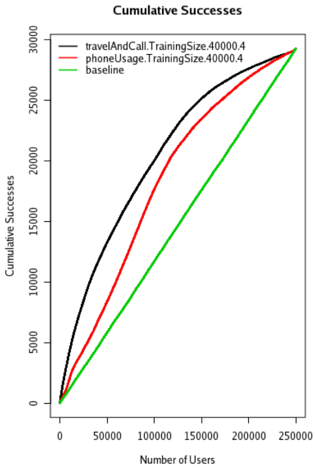
- **AUC ROC = area under the ROC curve**
- Often just referred to as "**AUC**"
- A single number commonly used to summarize classifier performance.
- Much more can be said about AUC and ROC curves...
- People also consider **AUC PR = area under the PR curve**

# Recall: The Cell Phone Churn Problem

- Cell phone customers often switch carriers. Called "churn".
- Often cheaper to retain a customer than to acquire a new one.
- You can try to retain a customer by giving a promotion, such as a discount.
- If you give a discount to somebody who was going to churn, you probably saved money.
- If you give a discount to somebody who was NOT going to churn, you wasted money.

- Now we've trained a classifier to predict churn.
- We need to choose a threshold on our score function
    - We will give a promotion to everybody with score above threshold.

# Lift Curves for Predicting Churners




- $x$ value: number of users targeted
- $y$ value is number churners in target group.
- Baseline is for a random score function
- Each curve is a **lift curve**
  - shows increase in successes from model over baseline