Lecture 3 White Board Notes:

**Performance Evaluation Review:**

Why 66/34 train-test split?

* Empirical testing found that 66/34 split was optimal for large datasets

Imagine 66/34 train-test split for for represents an independent experiment of the same dataset

* If dataset is small, there will be overlap in the datasets
  + Use cross validation

Probabilistic Classifiers

* Can transform into a deterministic classifier by setting a threshold e.g.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Probabilities** | | **Deterministic Prediction** | | **Actual** | |
| **Instance** |  |  |  |  |  |  |
| 1 | 0.8 | 0.2 | Y | N | 1 | 0 |

* Loss Functions:
  + 0\_1 Loss Function
    - Transform to deterministic
  + Quadratic Loss Function:
    - Bounds [0,2]
  + Informational Loss: , = predicted probability of the actual class
    - Bounds [0, )

Ex:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Instance** | **Predicted Probabilities** | | **Actual Probabilities** | | **Loss** | | |
|  |  |  |  | **0-1 Loss** | **Quadratic Loss** | **Information Loss** |
| 1 | .7 | .3 | 1 | 0 | 0 | .18 |  |
| 2 | .9 | .1 | 1 | 0 | 0 | .02 |  |
| 3 | .4 | .6 | 0 | 1 | 0 | .32 | . |
| 4 | .2 | .8 | 1 | 0 | 1 | 1.28 | . |
| 5 | .3 | .7 | 0 | 1 | 0 | .18 | . |
| 6 | .8 | .2 | 1 | 0 | 0 | .08 | . |

**Cost Sensitive & Evaluation Learning:**

“Everything has a cost...”

Ex1:

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Predicted | |
| Yes | No |
| Actual | Yes | 100 | 3 |
| No | 10 | 140 |
| Confusion Matrix | | Predicted | |
| Yes | No |
| Actual | Yes | TP | FN |
| No | FP | TN |

Ex2: Cost Calculation

|  |  |  |  |
| --- | --- | --- | --- |
| Cost Matrix |  | Predicted Class | |
| + | - |
| Actual Class | + | -1 | 100 |
| - | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| Model |  | Predicted Class | |
| + | - |
| Actual Class | + | 150 | 40 |
| - | 60 | 250 |

🡨 Typo in the lecture notes

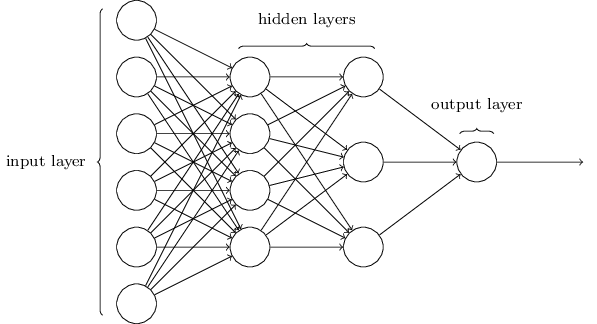
* Something to consider:
  + Choose higher accuracy w/ higher cost or lower accuracy w/ lower cost

Kappa-Statistic

* Compares an algorithm to a “random” algorithm
  + Better than random?
* To compare performance based on misclassification matrices
  + Historical Usage (time before computers/slow computers)
    - Model 1 vs Random Classifier
    - Model 1 vs Human
    - Human vs Human

Cost Sensitive Learning

* Modify Inputs
  + resampling of instances
    - 🡪 , is more costly
    - Create more negative examples
      * Oversampling
        + Replace each negative instance w/ exact same instances
        + SMOTE
        + K-NN
        + Deep Learning
    - Create less positive examples
      * Undersampling
        + Ex:1000 C1, 500 C2 🡪 500 C1, 500 C2
    - Combine oversampling and undersampling
      * Ex: 1000 C1, 100 C2 🡪 500 C1, 500 C2
  + Weighting of Instances
    - Assigns weight to each instance
* Modifying the learning algorithms
  + Cost sensitive boosting
  + Neural Networks



Lift/Gain Charts

Ex:

|  |  |  |
| --- | --- | --- |
|  | House hold | Respondents |
| Actual | 1,000,000 | 1000 |
| Data Mining 1 | 100,000 | 400 |
| Data Mining 2 | 400,000 | 800 |

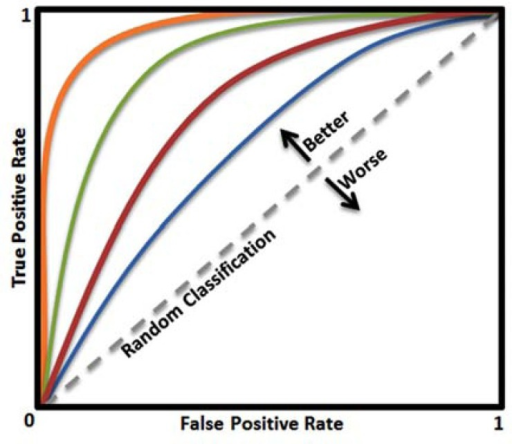
Actual 🡪 DM1:

Actual 🡪 DM2:

* Can plot results (see lecture notes)

ROC: Receiver Operating Characteristic

Sensitivity



1 - Specificity

* Each point in the line represents a certain threshold
* Each color represents different models/classifiers
* Whichever threshold gives perpendicularity closest to (top left corner) is the “best” for the given parameters
* Ex: How to plot?
  + Threshold = 0.5

|  |  |  |  |
| --- | --- | --- | --- |
| Model 1 | | Predicted | |
| C1 | C2 |
| Actual | C1 | 80 | 3 |
| C2 | 7 | 3 |

* This would be a point in the ROC curve
* Now try with different thresholds

AUC (Area Under Curve)

* Whichever ROC has more area is better

Continuous Outcomes:

* So far, above performance evaluations/metrics are for discrete outcomes
* MSE =
  + Can also use or
* Correlation Coefficient
  + Close to 1 means exact correlation

Which Model to Choose e.g. 1000 rules 🡪 99% Accuracy or 100 rules 🡪 97.6% Accuracy?

* Recall Occam’s Razor
* Is the 2.3% difference significant?

**Ensemble Learning (Overview)**

* Independently
  + Bootstrapping
  + Bagging
  + Random infusion
    - Random Forest
  + Rotation Forest
* Coordinated/Constructed
  + Boosting
  + Stacking