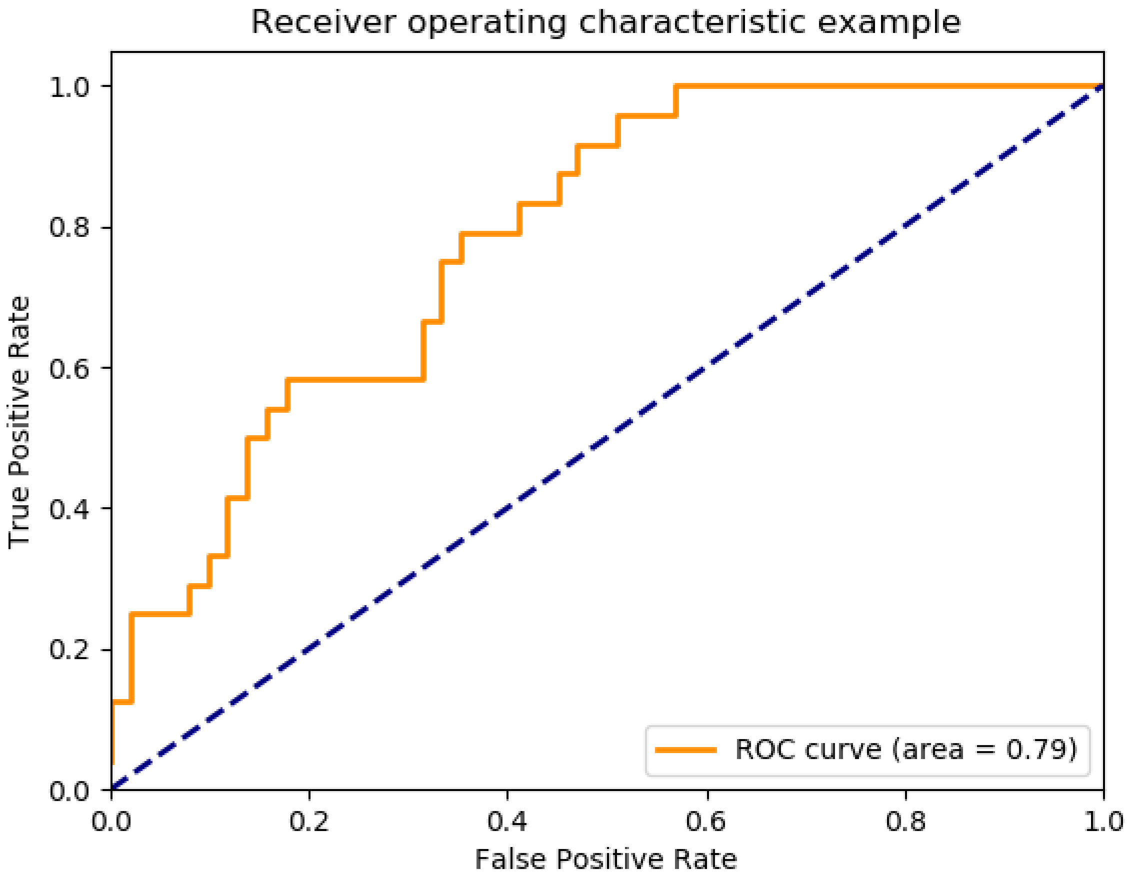
Week 7: Performance metrics



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

Week 6 review: Clustering

- What type of machine learning is this (supervised, unsupervised, reinforcement)?
  -
- What is the typical kmeans algorithm?
  -
  -
  -
  -
  -
- What can we do to reduce runtime if our data has too many features?
  -
- How is HCA different from kmeans?
  -
  -
  -

Week 6 review: Clustering

- What type of machine learning is this (supervised, unsupervised, reinforcement)?
  - unsupervised
- What is the typical kmeans algorithm?
  - 1. Pick cluster centers (kmeans++)
  - 2. Calculate distance from all points to all cluster centers
  - 3. Assign points to nearest cluster
  - 4. Calculate centroid of each cluster
  - Repeat 3 and 4 until assignments don't change
- What can we do to reduce runtime if our data has too many features?
  - Use PCA to reduce dimensions of features
- How is HCA different from kmeans?
  - HCA starts either as individual points or as one big cluster (agglomerative or divisive)
  - With agglomerative, uses point-point or cluster-cluster distances to group points/clusters into bigger clusters
  - 3 linkages: single, complete, average

Week 6 review quiz

- Cluster the data from auto.dt.nona.csv (week 2 content / data)
- Find the optimal number of clusters with the elbow plot and explain the scree plot
- Plot the mpg vs displacement, and color points by cluster
- If you have time, calculate some summary stats of the different clusters and compare/discuss
- Also could try scaling the data and comparing clusters to un-scaled
- Drop assignment under assignments in “Week 6 review quiz: K-means clustering” folder, a .R file is sufficient

Regression performance metrics

- What have we used so far?

Regression performance metrics

- What have we used so far?
  - Mainly root-mean-square error (RMSE) and R^2, which are good

$$RMSE = \sqrt{\frac{\sum_{i}^{n} \left(y_i - \hat{y}_i \right)^2}{n}}$$

$$R^2 = 1 - SSE/SST = 1 - \frac{\sum_{i}^{n} \left(y_i - \hat{y}_i \right)^2}{\sum_{i}^{n} \left(y_i - \bar{y} \right)^2}$$

Regression metrics

- Mean absolute error (MAE)
- Mean absolute percentage error (MAPE):
- Root-mean-square-log error (RMSLE)
- Can take other transforms of RMSE or MAE
  - e.g. inverse hyperbolic tangent (arctan); like log but can handle 0 and negative values

$$MAE = \frac{\sum_i |y_i - \hat{y}_i|}{n}$$

$$MAPE = \frac{\sum_i |(y_i - \hat{y}_i) / y_i|}{n}$$

$$RMSLE = \sqrt{\frac{\sum_i^n \left(\log(y_i + 1) - \log(\hat{y}_i + 1)\right)^2}{n}}$$

Classification performance metrics

- What have we used so far?

Classification performance metrics

- What have we used so far?
  - Accuracy (ok)
  - Confusion matrix

Other classification metrics

- Recall
- Precision
- F1
- Log loss
  - N = number of samples
  - M = number of classes
  - y_{ij} = binary indicator of class label
  - p_{ij} = model probability of class j for sample I
  - What are some problems with log loss?

$$logloss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log\left(\hat{p}_{ij}\right)$$

http://www.exegetic.biz/blog/2015/12/making-sense-logarithmic-loss/

Other classification metrics

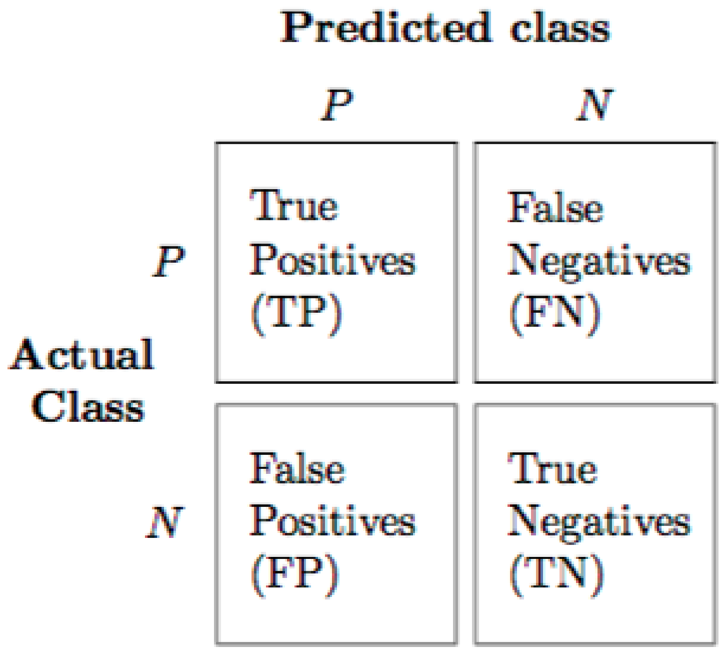
- Recall
- Precision
- F1
- Log loss
  - N = number of samples
  - M = number of classes
  - y_{ij} = binary indicator of class label
  - p_{ij} = model probability of class j for sample I
  - What are some problems with log loss?
    - No differentiation between false positives/false negatives
    - High penalty for low probability of correct class, highly non-linear penalty with increasing probability
    - No penalty for class label (y_{ij}) of 0, even if model is predicting significant probability for that class

$$logloss = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(\hat{p}_{ij})$$

http://www.exegetic.biz/blog/2015/12/making-sense-logarithmic-loss/

Confusion matrix

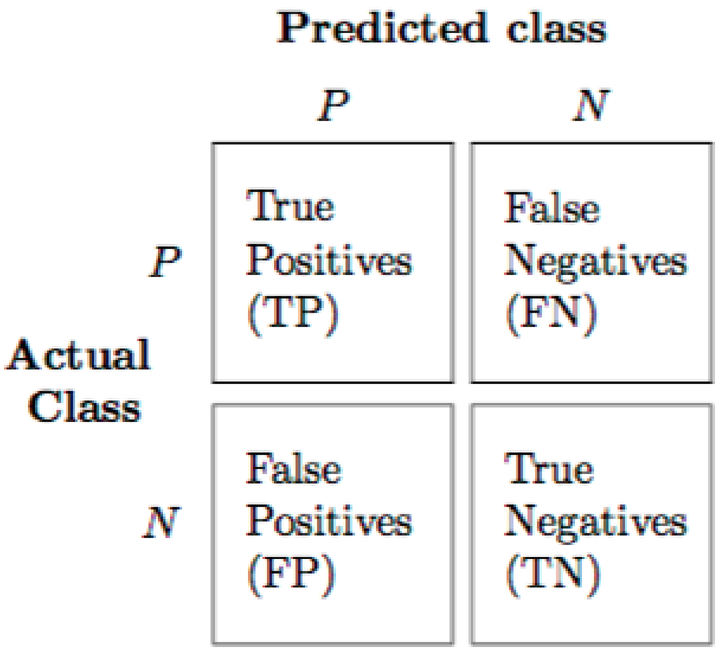
- When do we want to avoid
  - false negatives?
  - false positive?



https://rasbt.github.io/mlxtend/user_guide/evaluate/confusion_matrix/

Confusion matrix

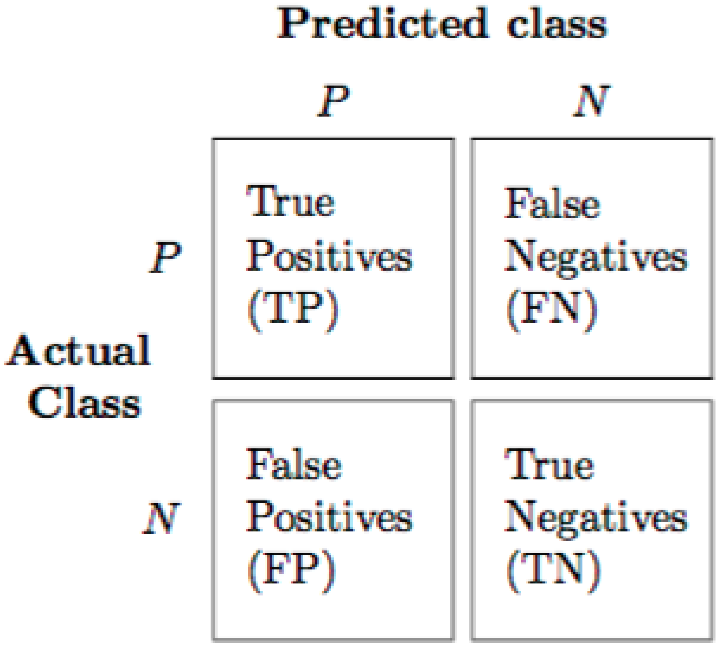
- When do we want to avoid
  - false negatives?
    - Cancer (or other disease) detection
  - false positive?
    - Marketing
    - Harsh treatments
      - e.g. chemotherapy for cance



https://rasbt.github.io/mlxtend/user_guide/evaluate/confusion_matrix/

Recall/precision/F1

- Recall – TP / (TP + FN)
  - TP / all actual positives
- Precision – TP / (TP + FP)
  - TP / all predicted positive
- F1 = 2 * Pr * Re / (Pr + Re)



https://rasbt.github.io/mlxtend/user_guide/evaluate/confusion_matrix/

Clustering performance metrics

- What did we use for determining goodness of kmeans clustering?

Clustering performance metrics

- What did we use for determining goodness of kmeans clustering?
  - Within sum of squares (WSS)
  - Silhouette score

What are models we can use for regression?

What are models we can use for regression?

- Linear/polynomial fits
- KNN
- Forest methods (random forest, xgboost, etc)
- Neural networks
- SVMs

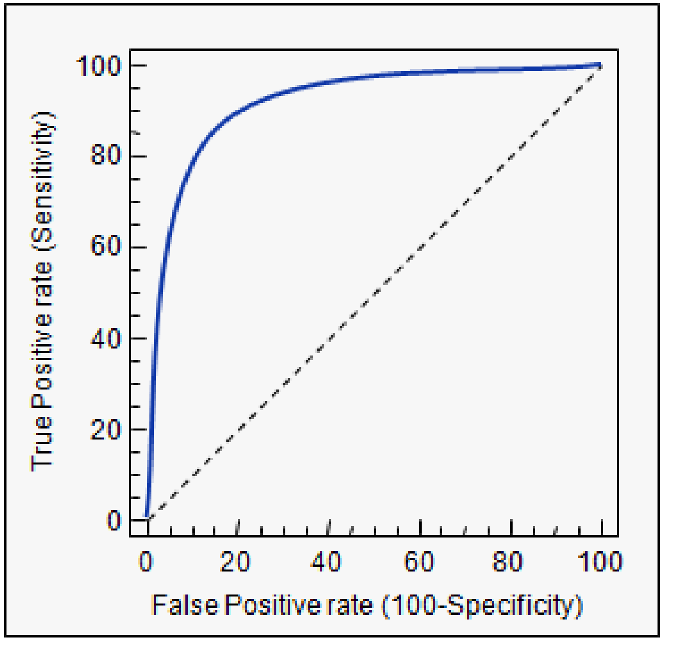
Models for classification?

Models for classification?

- KNN
- Logistic regression
- Forest models (random forest, xgboost, etc)
- Neural nets
- SVMs
- Naive Bayes

ROC curve & AUC

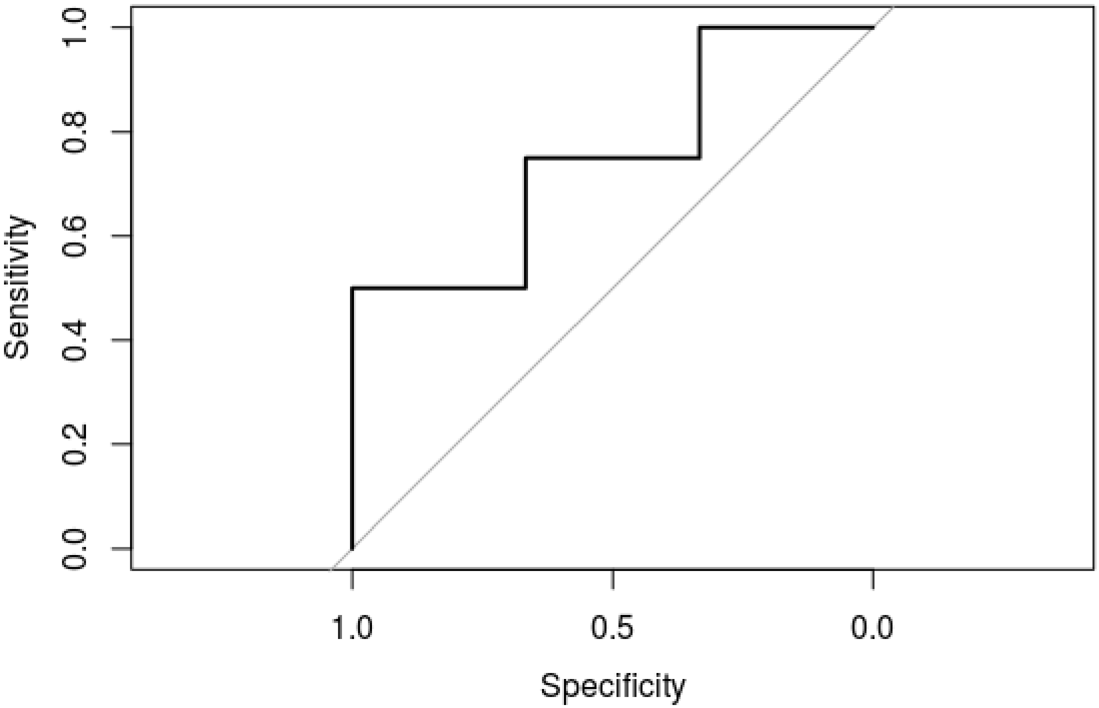
- ROC – receiver operating characteristic
- AUC – area under the curve
  - 1 means perfect classification, random guessing is the diagonal line (area of 0.5).



ROC curve & AUC

- Generate ROC curve by taking every prediction and setting threshold equal to that value
- roc.auc.demo.R under week 7

Predicted	Actual
0.15	0
0.23	1
0.45	0
0.55	1
0.67	0
0.88	1
0.97	1

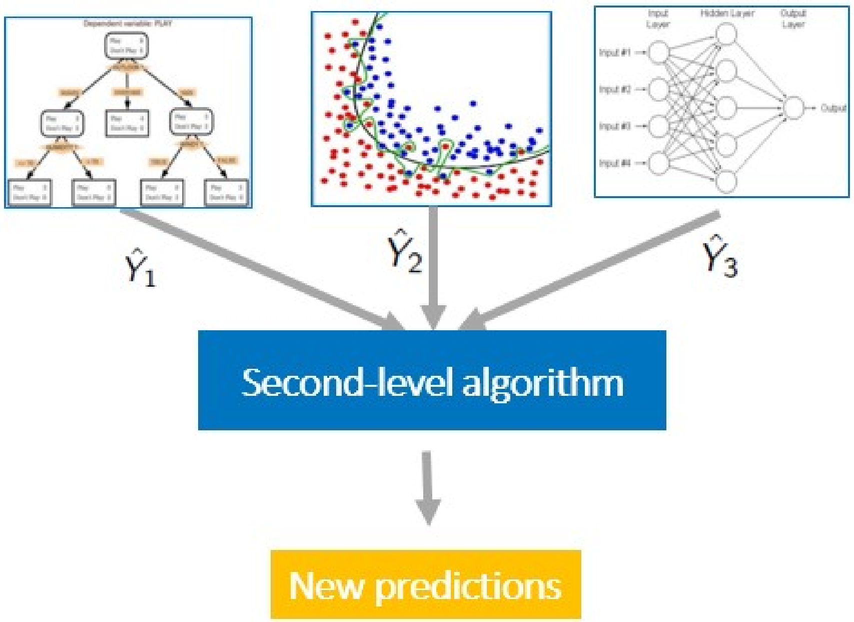


Comparing models

- Train on the same training set, evaluate on the same testing set
- Alternatively (probably better), use cross-validation, and train/evaluate models on the same splits
- Use the same metrics to compare them, of course
- Pick a metric that makes sense
  - For detecting a disease, maybe recall matters most, because we always want to be able to detect the disease. So optimize the model choice for that.

Ensembling models

- We can combine predictions from models to get an averaged prediction
  - Less bias like with random forests vs decision trees
- We could also take the predictions from multiple models, and feed them into another model...and do this with lots of variations
- Related: mixture of experts: https://en.wikipedia.org/wiki/Mixture_of_experts



https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions/

Demo (heart disease dataset)

- model.comparison.demo.R

- Compare different models' performance on the *same* train/test splits
- Procedure (one way to do it):
  - Optimize model hyperparameters with cross-validation
  - Fit model to full train set
  - Score/compare models on test set
    - Get accuracy, ROC curves/AUC score
    - Plot ROC curves
- Ensembling – averaging models

Neural Nets in R

- Keras also available in R:
  - https://keras.rstudio.com/
- No major difference in performance expected between R and Python for this, because Keras should be using C libraries to do the computations.

Exercise (pair)

- Try 3 different types of models to make classification predictions on the bank marketing dataset (in week 4 content)
  - Try some different hyperparameters in an effort to get the best predictions from each model
  - Don’t use more than 3 models, because fitting the other models is the individual part of the assignment
- Calculate the AUC score as a comparison, and at least 2 other metrics (could be accuracy, F1 score, precision, recall, etc)
  - Compare the models’ performance on the same train/test splits
- Plot the ROC curve from the best model, choose what you think the best threshold value is for making predictions
- Discuss the results in a .Rmd and post the .Rmd/exported PDF to the week 7 discussion

Rest of exercise (solo)

- Use at least 2 other models (that you haven’t tried yet) on the dataset and compare to your existing results.
  - Try some different hyperparameters for these to try to get the best result
- Ensemble (average) some or all of the models
  - You can also train a model to take in the predictions of all the other models and output a final prediction
- Post as as response to your first post
- Optional extra: make a profit curve and find the optimum prediction threshold and model with the profit curve
  - Assume a profit of $4 for a TP and -$5 for a FP as in the book example here (pg 212):
  - https://drive.google.com/file/d/0B1cm3fV8cnJwNDJFNmx2a2RBaTg/view
  - Also assume unlimited resources (no cost constraint)
    - If there was a cost constraint (choose a value), what would the optimum model and profit be?