Question 1: Explain ROC curves

Receiver Operator Characteristic (ROC) curves show the trade-off between False-Positive Rates (FPR) and True-Positive Rates (TPR) of a binary classification model as a function of varying the discrimination threshold. The use case is to evaluate the performance of a binary classifier. The ROC curve can conveniently show the results of many confusion matrices in a single chart where TPR is on the y-axis and FPR is on the x-axis. A 'good' ROC curve 'pushes toward the top-left quadrant of the chart.

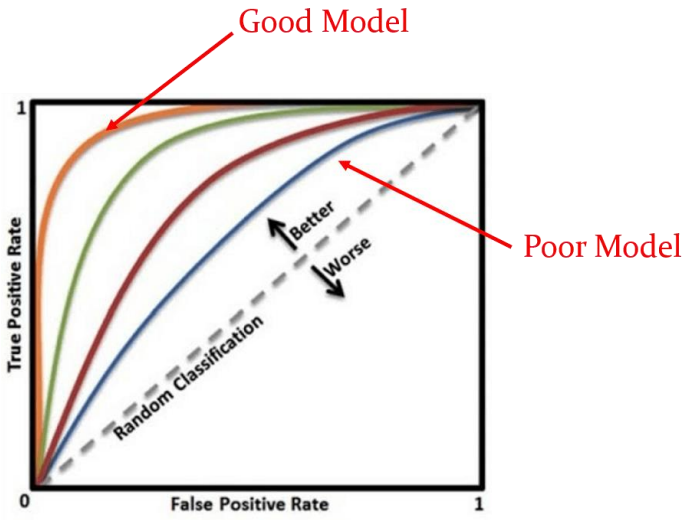

Figure 1: Typical ROC curves. Source: newtechdojo.com

The procedure for evaluating a ROC curve to outlined below.

Inputs required:

1. Logistic regression model from a binary classifier under test
2. Labelled test data (i.e. truth).

The analysis procedure:

1. Choose a classification threshold, which is a probability (0, 1).
2. Create confusion matrix: count false positives, true positives, false negatives, true negatives, and evaluate their rates.
3. Choose another classification threshold, repeat previous two steps.
4. Plot FPR and TPR for each threshold.

References:

- StatQuest with Josh Starmer, 'ROC and AUC, Clearly Explained!', https://www.youtube.com/watch?v=4jRBRDbJemM&ab_channel=StatQuestwithJoshStarmer
- Neptune AI, 'F1 Score vs. ROC AUC vs. Accuracy vs. PR AUC: Which Evaluation Metric Should You Choose?', https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc
- New Tech Dojo, ROC curve, https://www.newtechdojo.com/wp-content/uploads/2020/06/AUC-Pic6.png

Question 2: Describe cross-entropy and under what circumstances can it be used

Cross-entropy is a way to evaluate an error metric for a supervised learning binary classification model. When cross-entropy is aggregated it forms the result of the objective function (which is more commonly referred to as the loss function) of the optimisation algorithm used in neural networks, which relies on gradient descent techniques. Cross-entropy employs a logarithmic function to more strongly penalise errors in prediction of the model vs. 'truth'. Cross-entropy is sometimes known as 'log loss function'.

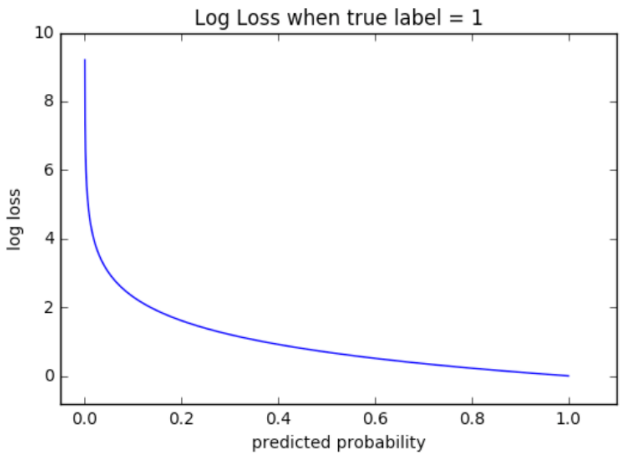

Figure 2: Log loss vs. predicted probability. Note the high loss as probability approaches zero.

Why is cross-entropy used?

- Cross-entropy improves optimiser performance. An objective function (loss) function based on any variant of sum of the square of residuals doesn't adequately penalise the optimiser when residuals are large.
- The gradient of the objective function, which governs the optimiser's behaviour, when using cross-entropy is significantly larger vs. square of residuals (or some variant of squaring errors).
- So 'cross-entropy helps take a relatively large step towards a better prediction' at the next optimiser step.

When is it used?

- Cross-entropy is typically used in supervised learning binary classification problems, where the problem is to determine whether an object is or isn't a class/category.

When is it not used?

- Supervised learning regression problems on the other hand, where predication is a continuous real number, typically use mean squared error or mean absolute error and therefore don't employ cross-entropy.
- Supervised learning multi-class classification problems typically employ soft-max cross-entropy, which is a 'normalise exponential function', and typically don't employ cross-entropy.

References:

- StatQuest with Jos Starmer, 'Neural Networks Part 6: Cross Entropy', https://www.youtube.com/watch?v=6ArSys5qHAU&t=303s&ab_channel=StatQuestwithJoshStarmer

- Normalized Nerd, 'Why do we need Cross Entropy Loss? (Visualised)', https://www.youtube.com/watch?v=gIx974WtVb4&ab_channel=NormalizedNerd
- JD Hao, 'Why cross entropy in classification', https://jdhao.github.io/2021/10/16/why_cross_entropy_in_classification/
- Numpy Ninja, 'Loss Functions - when to use which one?' https://www.numpyninja.com/post/loss-functions-when-to-use-which-one
- Super Data Science Team, 'CNN: Softmax vs. Cross-Entropy', https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-softmax-crossentropy
- Geoffrey Hinton, 'The softmax output function', https://www.youtube.com/watch?v=PHP8beSz5o4&ab_channel=ArtificialIntelligence-AllinOne
- ML Cheat Sheet, Cross-Entropy Curve, https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html

Question 3: Explain the similarities and differences of L1 loss and MSE loss in training a classifier (2 marks).

A classifier is a supervised learning problem. In general, a classifier evaluates a cost function. L1 and MSE loss are two different formulations of a cost function.

What is L1 loss?

- L1 loss is also known as Mean Absolute Error (MAE) according to Pytorch.
- L1 loss evaluates the mean of the absolute differences between each 'truth' value (y_i) and its 'predicted' value (y_hat)) per the below, where n is the number of training samples.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Figure 3: L1 loss formula: Source: https://miro.medium.com/

What is MSE loss?

- Mean Squared Error (MSE) loss evaluates the mean/average of the sum of the squares of differences between 'predicted' vs. 'actual'.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

Figure 4: Mean Squared Error. Source: stackoverflow.com

What are the similarities?

- Broadly speaking, L1 loss and MSE loss both evaluate (in gross terms) the difference between the ‘truth’ value and the ‘predicted’ value and sum that for all the truth/predicted value pairs.

What are the differences?

- However, the formulas speak for themselves. L1 loss is simply the sum of all the absolute errors, while MSE is the sum of the square of the errors divided by the number of truth-predicted value pairs.
- Consequently, L1 loss is *'less sensitive to outliers but harder to optimise'*. MSE is less forgiving when errors are large due to the squaring term. Practically, this means if the model is more sensitive to outliers and large differences between ‘truth’ and predicted.

References:

- Pytorch, L1 Loss, https://pytorch.org/docs/stable/generated/torch.nn.L1Loss.html

- David Inouye, 'Loss Functions and Regularization', https://www.davidinouye.com/course/ece57000-fall-2020/lectures/loss-functions-and-regularization.pdf
- Data Course, 'Evaluation of Regression Models in SciKitLearn', https://www.datacourses.com/evaluation-of-regression-models-in-scikit-learn-846/
- Chase Dowling, 'L1, L2 Loss Functions and Regression', https://cpatdowling.github.io/notebooks/regression_2
- Lo, 'Differences between L1 and L2 as Loss Function and Regularization', https://www.chioka.in/differences-between-l1-and-l2-as-loss-function-and-regularization/