Predictive Data Science

How to Assess Predictions

Evaluation techniques and strategies for Supervised Learning

- Data preparation
  - Train-Test-Validation
  - Cross-Fold Validation
  - Unbalanced class distribution
- Model Evaluation – metrics and interpretation
  - Numeric predictions
  - Categorical predictions – class membership
    - Confusion Matrix
    - ROC Curves
    - Learning Curves

Preparing datasets

Overall approach

- **Always** evaluate predictions using data that the model has never seen.
- Predictions based on data the model was trained on are likely to be optimistic, at best.
  - Training data predictions will not demonstrate the model’s ability to **generalize**.
  - Models that cannot generalize are said to be **overfitting**.
    - We must recognize and take steps to reduce overfitting in models.

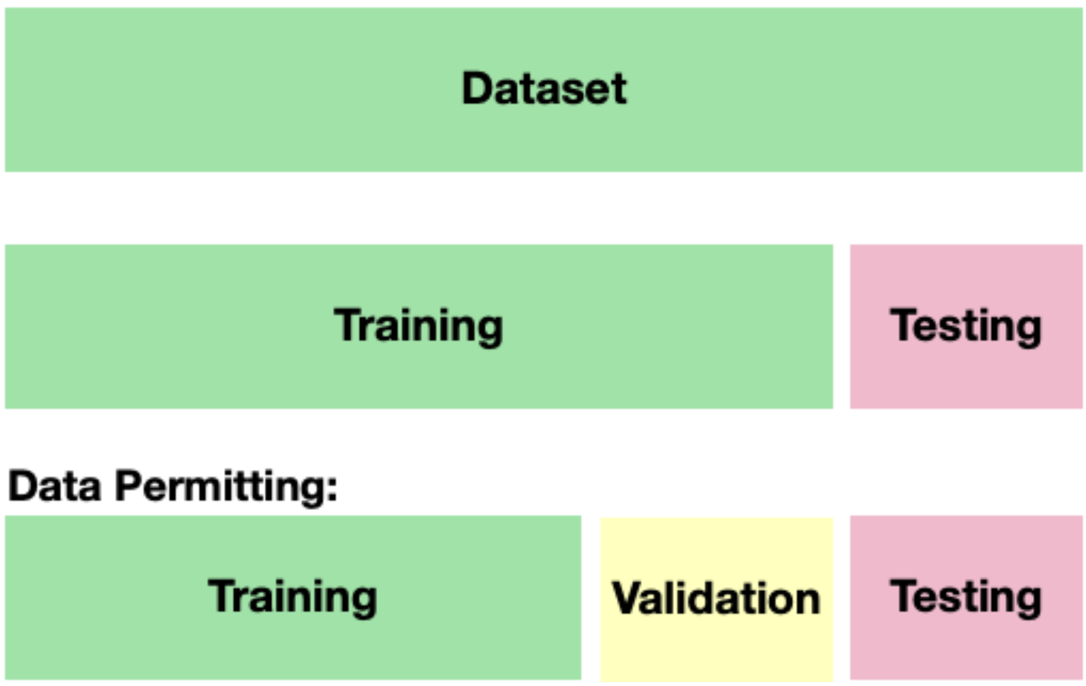
Train–Test Splits

- Before beginning training randomly sample the dataset and split the observations into a training set and a test set.
  - Most machine learning libraries provide methods for splitting datasets.
  - The most common splits are 70 : 30 or 80 : 20, with the smaller set reserved for testing.
  - Seed the randomizer with a constant value.

The test set must **never** be used to train the model.

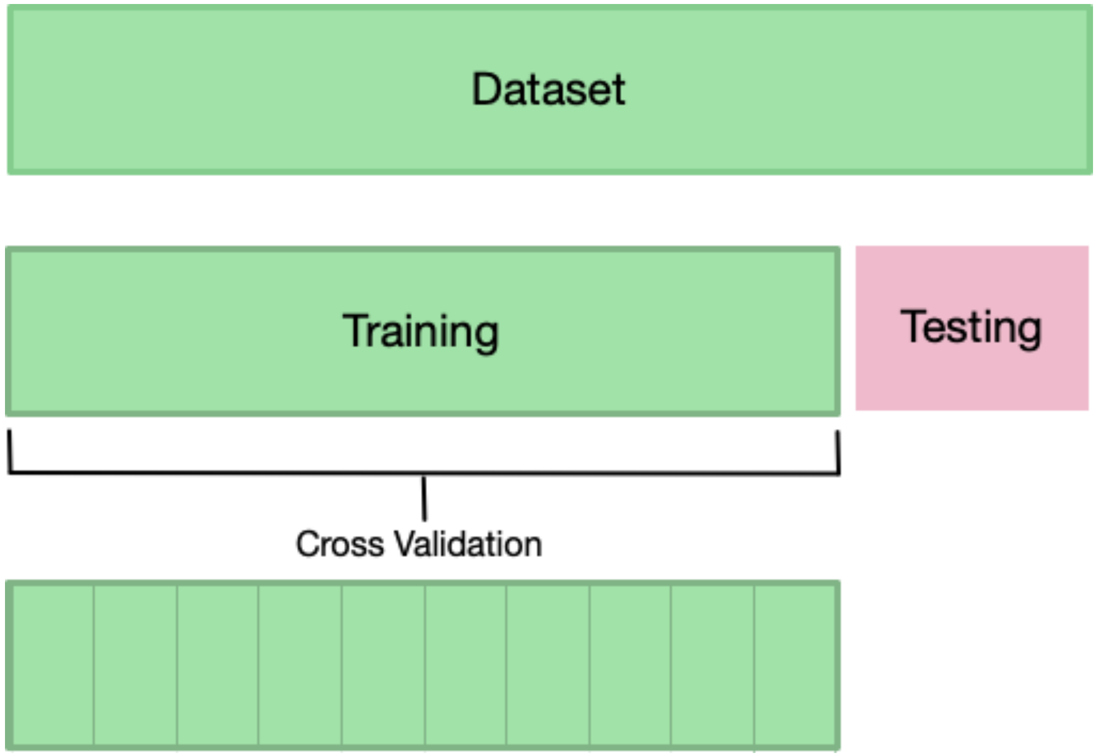
Train–Validation Splits

- Frequently, the training dataset is further split to form a validation set.
  - Validation set is used to select an algorithm or fine-tune the model – adjusting hyperparameters.
  - The Validation set is **not** used to train the model.



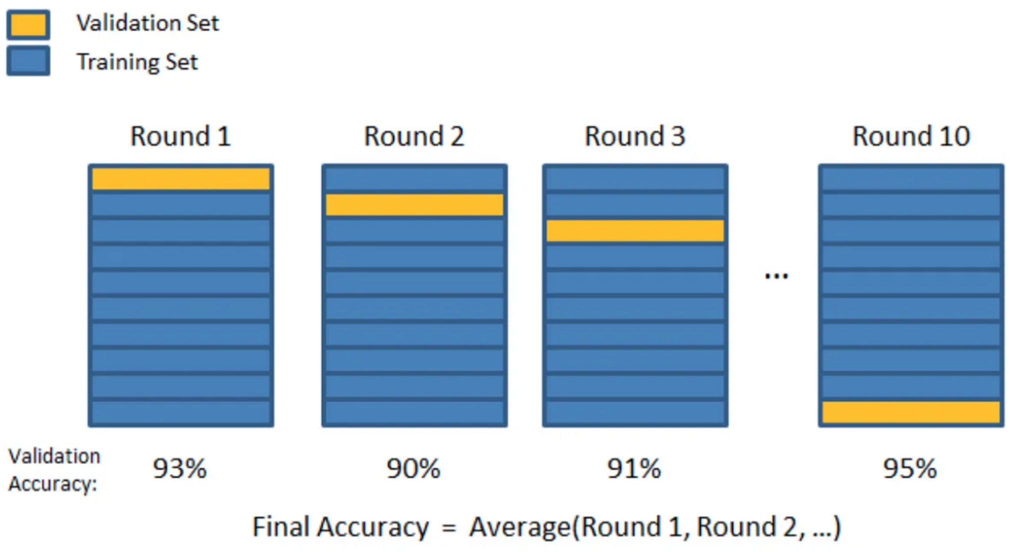
Cross Validation

- Split the Training set into *k* equal-sized *folds*.
  - *k* can be any by number up to the number of observations in the set.
  - *k* = 5 or 10 are the most common choices.



k-Fold Cross Validation

- Train the model *k* times.
- For each iteration, hold out a different one of the folds for use as the validation set.
- Average the results of all the iterations.
- Most machine learning libraries provide methods for cross-fold validation



Data Preparation Concerns for Classifiers

- The distribution of class occurrences can be unbalanced.
  - Fraudulent transactions vs. normal transactions; malignant tumors vs. benign
  - Distribution issues can be easily spotted during early analysis of the data set.
- Ideally, models will have a sufficient exposure to ‘minority’ classes.
- Training – Test – Validation splits run the risk of underrepresenting minority class(es) within some data partitions.
  - Minority class instances might even be excluded from partitions.

Dealing with Unbalanced Class Distribution

- Gather / Label the data set with class distribution in mind
- *Stratified Sampling*
  - Guarantees that each class is properly represented in both the training and test sets
- *Downsampling* – remove instances of majority classes
  - Can result in the loss of important information, especially with small data sets
- *Upsampling* – duplicate instances of the minority classes
  - Can lead to overfitting

Evaluating Numeric Predictions

Common Cost Functions

Mean-squared error: $\frac{\sum (p_i - a_i)^2}{n}$

Root mean-squared error: $\sqrt{\frac{\sum (p_i - a_i)^2}{n}}$

Mean absolute error: $\frac{\sum |p_i - a_i|}{n}$

Relative squared error: $\frac{\sum (p_i - a_i)^2}{\sum (a_i - \bar{a})^2}$

Root relative squared error: $\sqrt{\frac{\sum (p_1 - a_1)^2}{\sum (a_i - \bar{a})^2}}$

Relative absolute error: $\frac{\sum |p_i - a_i|}{\sum |a_i - \bar{a}|}$

Correlation coefficient: $\frac{S_{PA}}{\sqrt{S_p S_A}}, where\ S_{PA} = \frac{\sum (p_i - \bar{p})(a_i - \bar{a})}{n-1},\ S_P = \frac{\sum (p_i - \bar{p})^2}{n-1},\ S_A = \frac{\sum (a_i - \bar{a})^2}{n-1}$

p – predicted value, **a –** actual value, **ā** – mean of actual values

Common Cost Functions

- *Mean-squared error* is the most commonly used measure
  - Tends to exaggerate the effect of outliers
  - Easiest to manipulate mathematically
  - The square root puts the value in the same dimensions as the predicted value
- *Mean absolute error*
  - Does not exaggerate the effect of outliers
- *Relative squared error* measures error relative to the simple predictor: $a - \bar{a}$
- *Correlation coefficient* measures the correlations between the *a*'s and the *p*'s
- The best measure depends on the situation and what we are trying to minimize.

R^2

$$R^2 = \frac{var(mean) - var(fit)}{var(mean)}, where\ var(mean) = \frac{\sum (\bar{a} - a_i)^2}{n}, var(fit) = \frac{\sum (p_1 - a_i)^2}{n}$$

- *Answers the question: What percentage of the variance in the dependent variable is explained by the independent variables collectively?*
- Very commonly used measure of the goodness-of-fit for linear regression
  - An easy-to-understand, normalized value
- Cautions for use:
  - Should examine residual plots for possible bias – consistent under-predicting and over-predicting data along the curve.
  - Low R^2 values can still be good if data has inherent large unexplained variance.
  - High R^2 values might not be good. Low noise data, bias and overfitting can inflate R^2.

Evaluating Categorical Predictions

Evaluating classification predictions

- *The top-line classifier performance measure is usually the error (or success) rate.*
- What percentage of predictions were correct?
  - Correct Predictions / Total Instances
  - ML models will call this **Accuracy**
- Must be interpreted with the balance (or imbalance) of the dataset in mind.

Costs of Errors

- *Frequently, the cost of an incorrect classification is of far greater importance than overall accuracy.*
  - Loan decisions: cost of lending to a defaulter is greater than the lost opportunity of lending to a credit-worthy customer.
  - Medical imaging: cost of failing to detect a malignant tumor is greater than the 'false alarm' of mis-classifying a benign growth.
- Classifier outcomes:
  - True positives (TP): Classifier predicts positive when sample is positive
  - False positives (FP): Classifier predicts positive when sample is negative
  - True negatives (TN): Classifier predicts negative when sample is negative
  - False negatives (FN): Classifier predicts negative when sample is positive

Evaluation Tool: Confusion Matrix

Predicted Class

		Positive	**Negative**		
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$	**a.k.a Recall**
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$	
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$	

Using the information in the Confusion Matrix, we can calculate:

$$TPR = \frac{TP}{Actual\ Positive} = \frac{TP}{TP + FN}$$

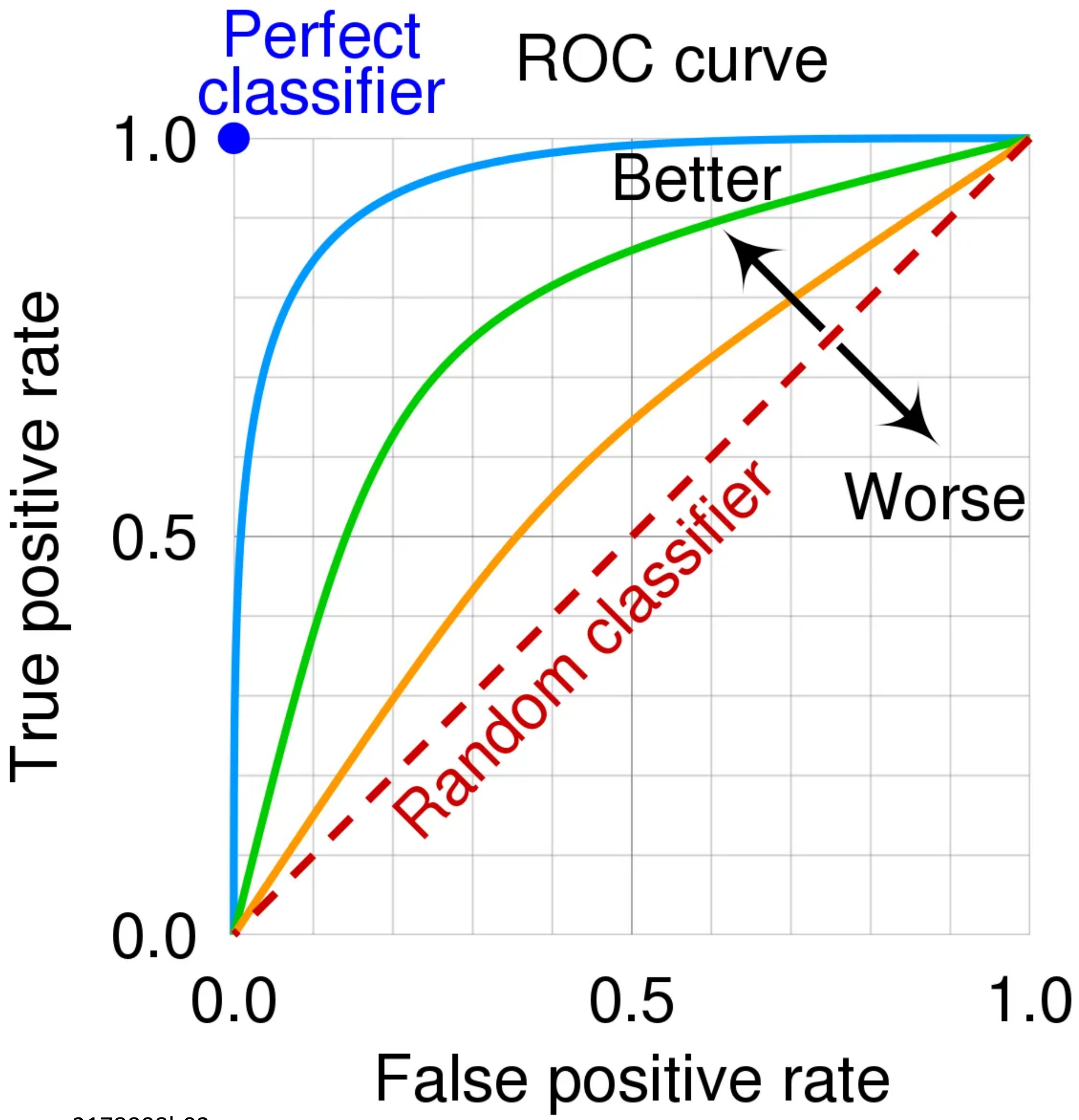
$$FNR = \frac{FN}{Actual\ Positive} = \frac{FN}{TP + FN}$$

$$TNR = \frac{TN}{Actual\ Negative} = \frac{TN}{TN + FP}$$

$$FPR = \frac{FP}{Actual\ Negative} = \frac{FP}{TN + FP}$$

Evaluation Tool: Receiver Operating Characteristic (ROC) Curves & Area Under the Curve (AUC)

- Some classifiers express predictions as the probability of class membership.
    - A threshold is set for the minimum probability for a positive prediction. Values below the threshold become negative predictions (for a binary classifier).
- An ROC curve shows the tradeoff between the True Positive Rate (TPR) and the False Positive Rate (FPR) as the threshold varies.
- The diagonal line from lower left to upper right represents a random classifier. The better performance is above that line and to the left.
- The AUC quantifies classifier performance with values between 0 and 1 – 1 being a perfect classifier.



Source: https://medium.com/@ilyurek/roc-curve-and-auc-evaluating-model-performance-c2178008b02

Evaluation Tool: Learning Curves

- Learning Curves are often used to analyze model performance under varying conditions.
- The graphs on the right show error rates for:
    - A given level of complexity
    - Both training data and test data
    - Different numbers of samples
- Observations:
    - As the number of samples increases, the error rate of the training set and the test set start to converge towards a value called the **bias**.
    - The model with greater complexity tends towards lower error rates.

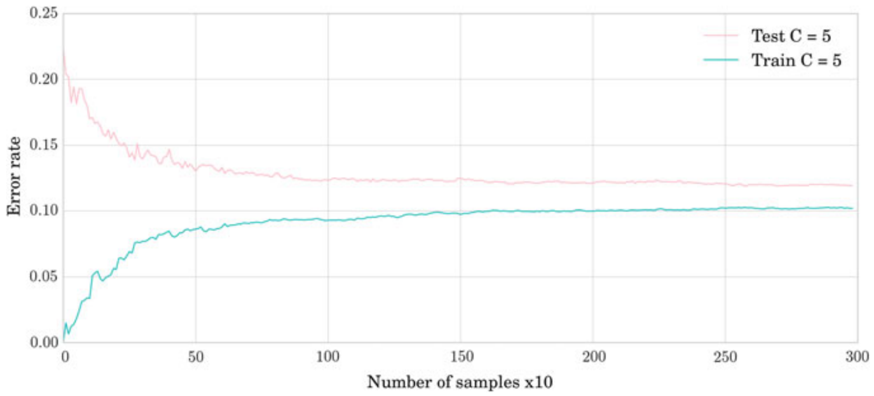


Fig. 1 Variable number of samples with complexity value of 5

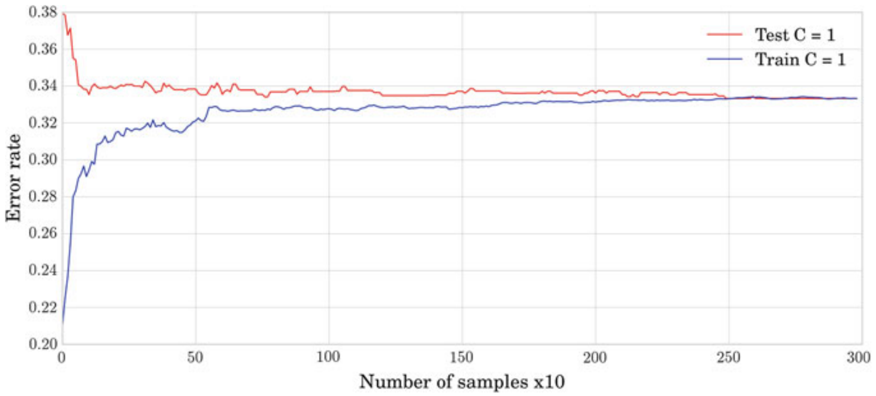


Fig. 2 Variable number of samples with complexity value of 1

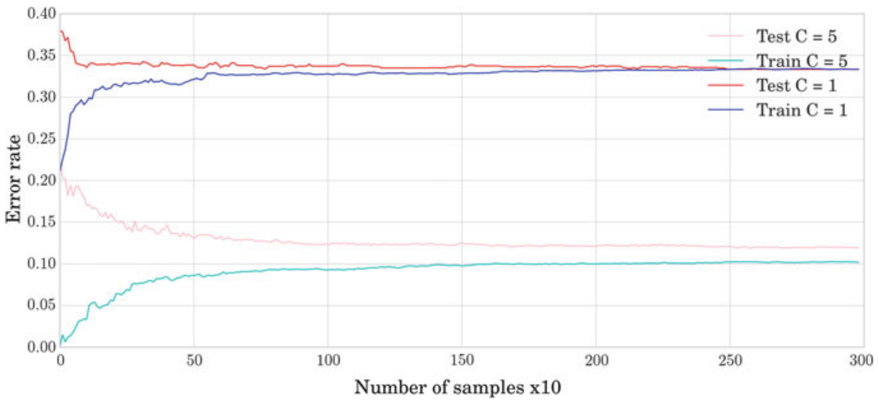


Fig. 3 Combined

Using the Learning Curve to Measure Underfitting / Overfitting

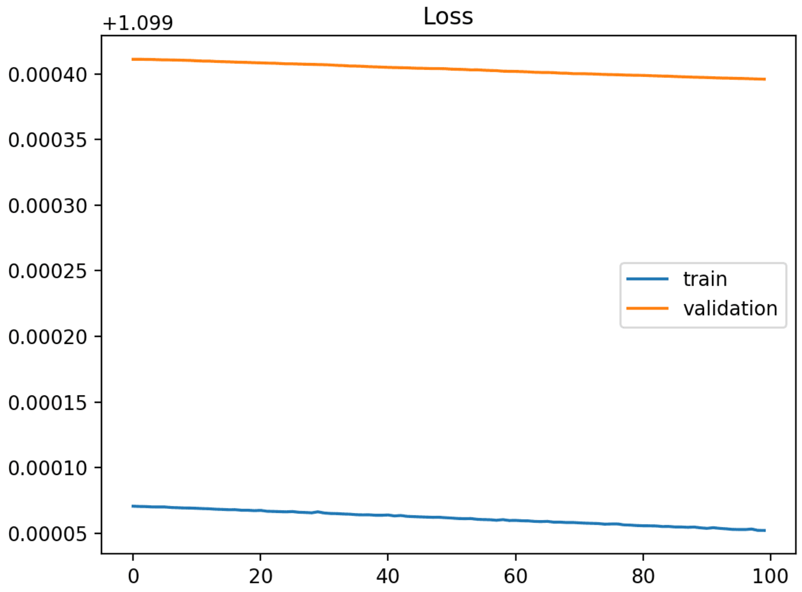


Fig. 1 A model that has failed to learn from the training dataset

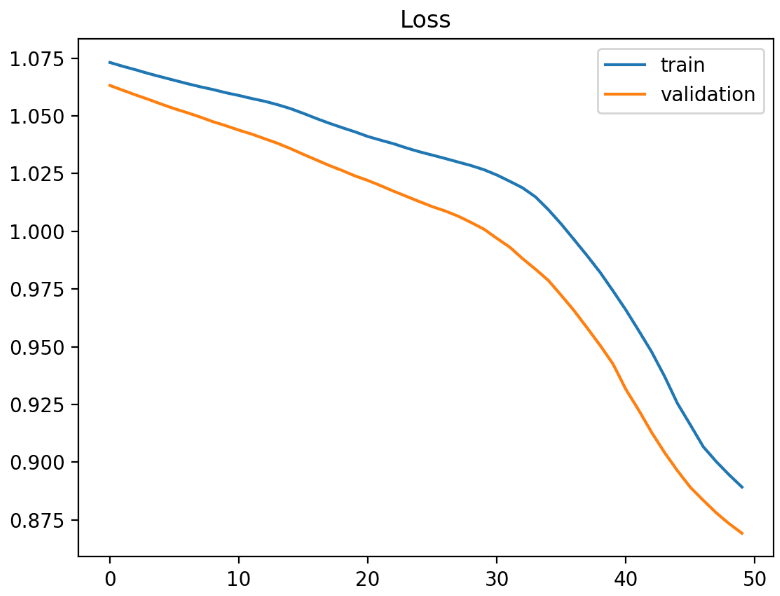


Fig. 2 A model that may be capable of further improvements

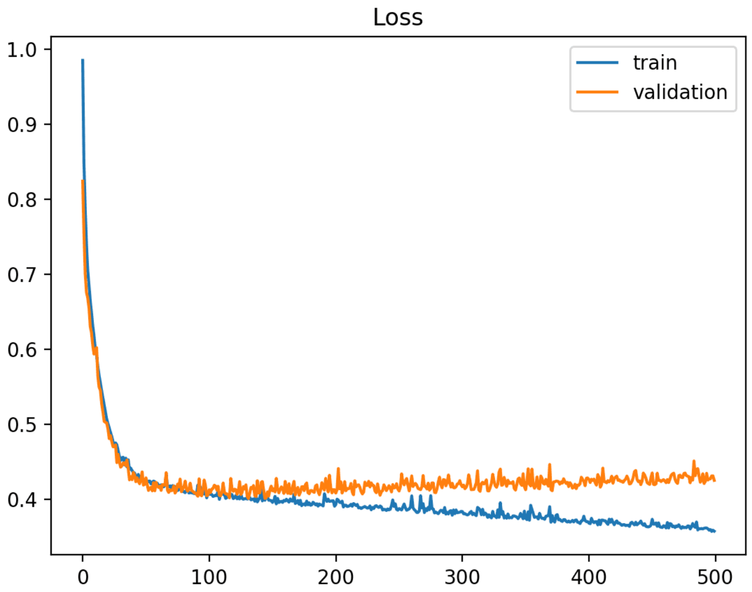


Fig. 3 A model that is overfitting

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/