राष्ट्रीय न्यायालयिक विज्ञान विश्वविद्यालय
(राष्ट्रीय महत्त्व का संस्थान, गृह मंत्रालय, भारत सरकार)
National Forensic Sciences University
(An Institution of National Importance under Ministry of Home Affairs, Government of India)

Datasets, Phases, Parametric Models & Measuring Performance

Datasets

- A dataset is nothing more than a collection of data.
- A dataset has a finite number of elements, and the ML algorithm will loop over this dataset several times, trying to understand the data structure, until it solves the task it is asked to address.
- Neural Networks and Deep Learning generally consider all the data at once, while other algorithms iteratively look at a small subset of the data at each training iteration.

Datasets

- If the same dataset is used during training and also to test the performance of algorithm, then one cannot decide whether the algorithm performs well on unseen data or not.
- The most common practice is to split the dataset into three parts: Training Set, Validation Set and Test Set.

Datasets

- Training set: The subset which is used to train the model.
- Validation set: The subset to measure the model's performance during the training and also to perform hyperparameter tuning/searches.
- Test set: The subset to never touch during the training or validation phases. This is used only to run the final performance evaluation.

Datasets

- The training set is usually the bigger subset since it must be a meaningful representation of the whole dataset.
- The validation and test sets are smaller and generally the same size—of course, this is just something general; there are no constraints about the dataset's cardinality.
- In fact, the only thing that matters is that they're big enough for the algorithm to be trained on and represented.

Datasets

- As a common practice the model learns from the training set, its performance during the training process is evaluated using the validation set, and the final performance evaluation is done on the test set: this allows one to correctly define and train supervised learning algorithms that could generalize well, and therefore work well even on unseen data.

Phases

- Supervised learning algorithms are two-phase algorithms.
- Given a supervised learning problem—let's say, a classification problem—the algorithm tries to solve it during the first phase, called the training phase, and its performance is measured in the second phase, called the testing phase.
- The first phase uses training and validation datasets and second phase uses testing dataset.

Phases

- Training and Validation Phase:
  - The algorithm analyzes the dataset to generate a theory that is valid for the data it has been trained on, and also for items it has never seen.
  - The algorithm, therefore, tries to discover and **generalize** a concept.
  - At the end of every training epoch, a performance evaluation using a metric on the validation set should be performed.

Phases

- Testing Phase:
  - The learned theory is applied to labeled examples that were never seen during the training and validation phases.
  - This allows us to test how the algorithm performs on data that has never been used to train or select the model hyperparameters—a real-life scenario.

Parametric Models

- A parametric model is a model that can be described using a function, where the input and output are known and the aim is to change the model parameters so that, given a particular input, the model produces the expected output.
- However, when using complex models (with a considerable number of adjustable parameters, as in the case of neural networks), adjusting the parameters can lead to undesired results.

Parametric Models

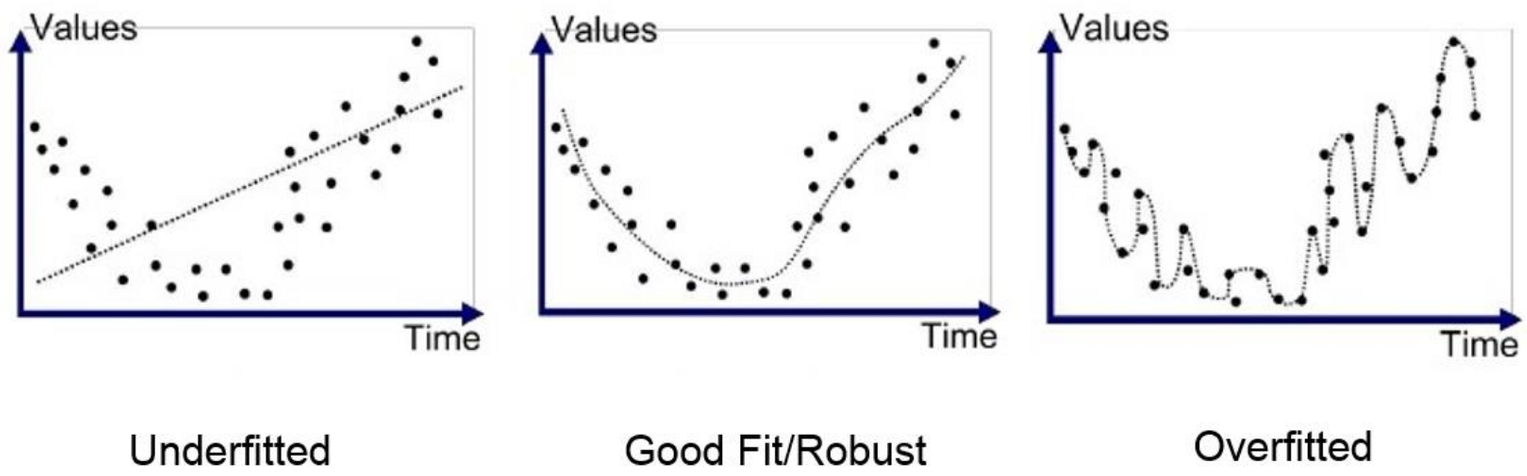
- If our model is composed of just two parameters and we are trying to model a linear phenomenon, there are no problems.
- But if we are trying to classify the dataset, we can't use a simple linear model since it is easy to see that the function we have to learn about to separate the different classes is not a simple line (Refer non-linearity concept).

Parametric Models

- The model is designed to adapt its parameters only to fit the training data, almost *memorizing* the dataset and thus losing every generalization capability.
- This pathological phenomenon is called **overfitting**, and it happens when we are using a model that's too complex to model a simple event.
- There's also an opposite scenario, called **underfitting**, that occurs when our model is too simple for the dataset and therefore is not able to capture all the complexity of the data.

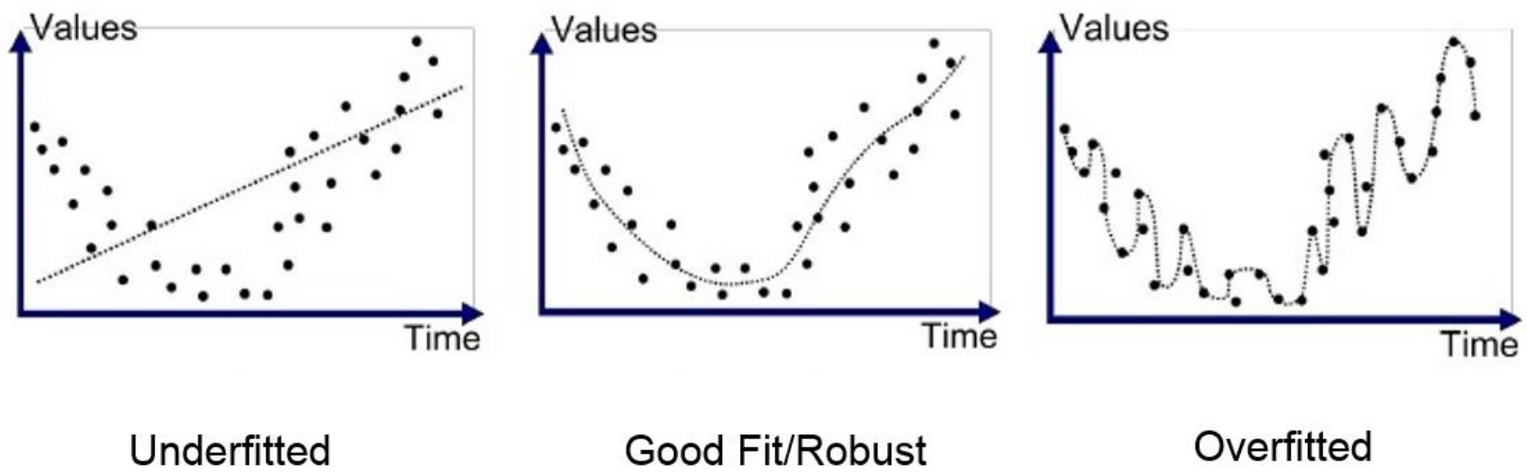
Parametric Models

- Every ML model aims to learn, and will adapt its parameters so that it's robust to noise and generalize, which means it tries to find a suitable approximate function representing the relationship between the predictors and the response. [Fig. Ref: Hands-On Neural Networks with TensorFlow 2.0 By Paolo Galeone]



Parametric Models

- Every ML model aims to learn, and will adapt its parameters so that it's robust to noise and generalize, which means it tries to find a suitable approximate function representing the relationship between the predictors and the response. [Fig. Ref: Hands-On Neural Networks with TensorFlow 2.0 By Paolo Galeone]



Measuring Performance

- Measuring the performance of a model is something that one can always do on every dataset split.
- During the training phase, one can measure the performance of the algorithm on the training set itself, as well as on the validation set.
- Plotting how the curves change during the training and analyzing the relationships between the validation and training curve allow to quickly identify overfitting and underfitting.

Measuring Performance

- Mean Absolute Error (MAE):
  - MAE is the average of the absolute difference between the original and the predicted values.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Measuring Performance

- MAE…
  - The MAE value has no upper bound, and its lower bound is 0.
  - It should be evident that we want the MAE value to be as close as possible to 0.
  - MAE gives us an indication of how far the predictions are from the actual output.

Measuring Performance

- Mean Squared Error (MSE)
  - The MSE is the average of the squared difference between the original and the predicted values.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Measuring Performance

- Mean Squared Error (MSE)…
  - Just like MAE, MSE has no upper bound and its lower bound is 0.
  - On the contrary, the presence of the square terms makes the metric less easy to interpret.
  - A good practice to follow is to consider both metrics so that you get as much information as possible about the distribution of the errors.
  - The RMSE is nothing but $\sqrt{MSE}$

Measuring Performance

- Using Accuracy:
  - Accuracy, the ratio of the number of correct predictions made to the number of all predictions made, is used to measure classification performance.
  - Remember that metrics can be used during the training phase to measure the model's performance, we can monitor how the training is going on by looking at the validation accuracy and the training accuracy to detect if our model is overfitting or underfitting the training data.

Measuring Performance

- Using Accuracy:
  - If the model is able to model the relationships present in the data, the training accuracy increases; if it doesn't, the model is too simple and we are underfitting the data.
  - If our training accuracy increases, we can start looking at the validation accuracy: if the validation accuracy stops growing or even starts decreasing, the model is overfitting the training data and we should stop the training.

Measuring Performance

- Using the Confusion Matrix:
  - The confusion matrix is a tabular way of representing a classifier's performance.
  - It can be used to summarize how the classifier behaved on the test set, and it can be used only in the case of multi-class classification problems.
  - Each row of the matrix represents the instances in a predicted class, while each column represents the instances in an actual class.

Measuring Performance

- Using the Confusion Matrix:
  - It is worth noting that the confusion matrix is not a metric; in fact, the matrix alone does not measure the model's performance, but is the basis for computing several useful metrics, all of them based on the concepts of true positives, true negatives, false positives, and false negatives.
  - These all terms refer to a single class. E.g., for a multiclass classification problem, whose classes are A, B, ..., Z, we can have following:

Measuring Performance

- Using the Confusion Matrix:
  - (TP) True positives of A: All A instances that are classified as A
  - (TN) True negatives of A: All non-A instances that are not classified as A
  - (FP) False positives of A: All non-A instances that are classified as A
  - (FN) False negatives of A: All A instances that are not classified as A

Measuring Performance

- Using the Confusion Matrix:

	Actual Positive	Actual Negative
Predicted Positive	True Positives (TP)	False Positives (FP)
Predicted Negative	False Negatives (FN)	True Negatives (TN)

Measuring Performance

- Using the Confusion Matrix:
  - This, of course, can be applied to every class in the dataset so that we get these four values for every class.
  - The most important metrics we can compute that have the TP, TN, FP, and FN values are precision, recall, and the F1 score.

Measuring Performance

- Precision:
  - Precision is the number of correct positives results, divided by the number of positive results predicted:

$$precision = \frac{TP}{TP + FP}$$

Measuring Performance

- Precision:
  - The metric name itself describes that we measure a number in the [0,1] range that indicates how accurate the predictions of the classifier are: the higher, the better.
  - High precision only means that, when we predict the positive class, we are precise in detecting it.
  - But this does not mean that we are also accurate when we're not detecting this class.

Measuring Performance

- Recall:
  - The recall is the number of correct positive results, divided by the number of all relevant samples (for example, all the samples that should be classified as positive):

$$recall = \frac{TP}{TP + FN}$$

Measuring Performance

- Recall:
  - Just like precision, recall is a number in the [0,1] range that indicates the percentage of correctly classified samples over all the samples of that class.
  - The recall is an important metric, especially in problems such as object detection in images.
  - Measuring the precision and recall of a binary classifier allows you to tune the classifier's performance, making it behave as needed.

Measuring Performance

- F1 Score:
  - The F1 score is the harmonic mean between precision and recall.
  - This number, which is in the [0,1] range, indicates how precise the classifier is (precision) and how robust it is (recall).
  - The greater the F1 score, the better the overall performance of the model:

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Measuring Performance

- Using the area under the ROC Curve:
  - The area under the Receiving Operating Characteristic (ROC) curve is one of the most used metrics for the evaluation of binary classification problems.
  - Most classifiers produce a score in the [0,1] range and not directly as a classification label. The score must be thresholded to decide the classification. A natural threshold is to classify it as positive when the score is higher than 0.5 and negative otherwise.

Measuring Performance

- Using the area under the ROC Curve:
  - The results of the threshold variations can be taken into account by plotting the ROC curve.
  - The ROC curve takes into account the false positive rate (specificity) and the true positive rate (sensitivity): binary classification problems are a trade-off between these two values.
  - We can describe these values as follows:

Measuring Performance

- Using the area under the ROC Curve:
  - Sensitivity: The true positive rate is defined as the proportion of positive data points that are correctly considered positive, with respect to all the positive data points:

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

Measuring Performance

- Using the area under the ROC Curve:
  - Specificity: The false positive rate (FPR) is defined as the proportion of negative data points that are considered positive, with respect to all the negative data points:

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

Measuring Performance

- Using the area under the ROC Curve:
  - The AUC is the area under the ROC curve, and is obtained by varying the classification threshold.
  - It is clear that both TPR and FPR have values in the [0,1] range, and the graph is drawn by varying the classification threshold of the classifier in order to get different pairs of TPR and FPR for every threshold value.
  - The AUC is in the [0,1] range too and the greater the value, the better the model is.

References

1. Hands-On Neural Networks with TensorFlow 2.0 By Paolo Galeone