Performance of a classifier

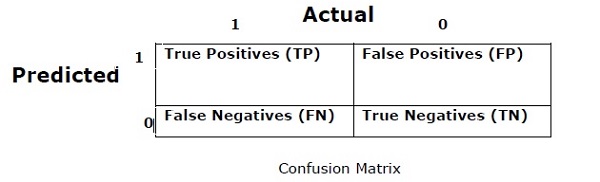
After implementing a machine learning algorithm, we need to find out how effective the model is.

The criteria for measuring the effectiveness may be based upon datasets and metric. For evaluating different machine learning algorithms, we can use different performance metrics.

For example, suppose if a classifier is used to distinguish between images of different objects, we can use the classification performance metrics such as average accuracy, AUC, etc. In one or other sense, the metric we choose to evaluate our machine learning model is very important because the choice of metrics influences how the performance of a machine learning algorithm is measured and compared. Following are some of the metrics −

Confusion Matrix

Basically it is used for classification problem where the output can be of two or more types of classes. It is the easiest way to measure the performance of a classifier. A confusion matrix is basically a table with two dimensions namely “Actual” and “Predicted”. Both the dimensions have “True Positives (TP)”, “True Negatives (TN)”, “False Positives (FP)”, “False Negatives (FN)”.



In the confusion matrix above, 1 is for positive class and 0 is for negative class.

Following are the terms associated with Confusion matrix −

* **True Positives −** TPs are the cases when the actual class of data point was 1 and the predicted is also 1.
* **True Negatives −** TNs are the cases when the actual class of the data point was 0 and the predicted is also 0.
* **False Positives −** FPs are the cases when the actual class of data point was 0 and the predicted is also 1.
* **False Negatives −** FNs are the cases when the actual class of the data point was 1 and the predicted is also 0.

Accuracy

The confusion matrix itself is not a performance measure as such but almost all the performance matrices are based on the confusion matrix. One of them is accuracy. In classification problems, it may be defined as the number of correct predictions made by the model over all kinds of predictions made. The formula for calculating the accuracy is as follows −

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

Precision

It is mostly used in document retrievals. It may be defined as how many of the returned documents are correct. Following is the formula for calculating the precision −

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

Recall or Sensitivity

It may be defined as how many of the positives do the model return. Following is the formula for calculating the recall/sensitivity of the model −

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

Specificity

It may be defined as how many of the negatives do the model return. It is exactly opposite to recall. Following is the formula for calculating the specificity of the model −

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

Class Imbalance Problem

Class imbalance is the scenario where the number of observations belonging to one class is significantly lower than those belonging to the other classes. For example, this problem is prominent in the scenario where we need to identify the rare diseases, fraudulent transactions in bank etc.

Example of imbalanced classes

Let us consider an example of fraud detection data set to understand the concept of imbalanced class −

Total observations = 5000

Fraudulent Observations = 50

Non-Fraudulent Observations = 4950

Event Rate = 1%

Solution

**Balancing the classes’** acts as a solution to imbalanced classes. The main objective of balancing the classes is to either increase the frequency of the minority class or decrease the frequency of the majority class. Following are the approaches to solve the issue of imbalances classes −

Re-Sampling

Re-sampling is a series of methods used to reconstruct the sample data sets − both training sets and testing sets. Re-sampling is done to improve the accuracy of model. Following are some re-sampling techniques −

* **Random Under-Sampling** − This technique aims to balance class distribution by randomly eliminating majority class examples. This is done until the majority and minority class instances are balanced out.

Total observations = 5000

Fraudulent Observations = 50

Non-Fraudulent Observations = 4950

Event Rate = 1%

In this case, we are taking 10% samples without replacement from non-fraud instances and then combine them with the fraud instances −

Non-fraudulent observations after random under sampling = 10% of 4950 = 495

Total observations after combining them with fraudulent observations = 50+495 = 545

Hence now, the event rate for new dataset after under sampling = 9%

The main advantage of this technique is that it can reduce run time and improve storage. But on the other side, it can discard useful information while reducing the number of training data samples.

* **Random Over-Sampling** − This technique aims to balance class distribution by increasing the number of instances in the minority class by replicating them.

Total observations = 5000

Fraudulent Observations = 50

Non-Fraudulent Observations = 4950

Event Rate = 1%

In case we are replicating 50 fraudulent observations 30 times then fraudulent observations after replicating the minority class observations would be 1500. And then total observations in the new data after oversampling would be 4950+1500 = 6450. Hence the event rate for the new data set would be 1500/6450 = 23%.

The main advantage of this method is that there would be no loss of useful information. But on the other hand, it has the increased chances of over-fitting because it replicates the minority class events.

Ensemble Techniques

This methodology basically is used to modify existing classification algorithms to make them appropriate for imbalanced data sets. In this approach we construct several two stage classifier from the original data and then aggregate their predictions. Random forest classifier is an example of ensemble based classifier.