DHRUV GARG 16 BCE 1190

MACHINE LEARNING - DIGITAL ASSIGNMENT 3

ANALYSIS OF METRICS FOR EVALUATION OF MODELS

INTRODUCTION

Before getting started with the implementation, we must first understand the need for using various metrics for model evaluation.

When we build machine learning models (in this case classification models), we must have a strong reasoning for choosing between different model types, tuning parameters and features. Thus, **model evaluation helps us to estimate how well a model would generalize** on out-of-sample data, ie. the data which it has not seen yet. To quantify the performance of various models and to compare them with one another, we have evaluation metrics.

There are 3 model evaluation procedures:

1. **Training and testing on same data**
   This is a bad evaluation procedure since it produces overly complex models that overfit the training data.
2. **Train / test split**
   Split the dataset into two pieces so that the model can be trained and tested on different data. This gives a better out-of-sample performance, but there is a drawback of high variance estimate.
3. **K-fold cross validation**
   Systematically create K train/test splits and average the results together. This gives even better estimate of out-of-sample performance. However it runs K times slower than train/test split.

Model evaluation metrics commonly used:

For regression: Mean squared error, root mean squared error, mean absolute error

For classification: Confusion matrix, accuracy, error, sensitivity, specificity, precision, ROC and AUC.

Since I have implemented classification, I will be describing the accuracy metrics for classification.

Classification accuracy: It tells us about the number of correct predictions made by the model. It is the most common and simplest measure to evaluate a classifier. Formula:

$$\text{Acc} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{c} f(i,j)C(i,j)}{m}$$

Null accuracy: It is the accuracy that can be achieved by a model that always predicts the most frequent class. This gives the accuracy measure that a basic model that always predicts the mode. It is a good way to know the minimum accuracy to be achieved with our model.

Confusion matrix: It gives the results of the predictions as a matrix. Each observation is represented in exactly one box.

	Class 1 Predicted	*Class 2 Predicted*
Class 1 Actual	**TP**	**FN**
Class 2 Actual	**FP**	**TN**

True Positives (TP): Observation is positive, and is predicted to be positive.

True negatives (TN): Observation is negative, and is predicted to be negative.

False positive (FP): Observation is negative, but is predicted positive.

False negative (FN): Observation is positive, but is predicted negative.

The confusion matrix is important since we can use it to derive numerous other accuracy measures.

Classification error: It gives us the total percentage of error by the model in classification. It can be found by:

Error = (FP + FN) / (FP + FN + TP + TN)

Sensitivity: It is a measure of the correctness of predictions when the true value is positive. This is a measure that we want to maximize. The formula is given as:

Sensitivity = TP / (FN + TP)

Specificity: It is a measure of the correctness of predictions when the true value is negative. This is a measure that we want to maximize. The formula is given as:

Specificity = TN / (TN + FP)

False positive rate:

It tells us the measure of the in-correctness of a prediction when the actual prediction is negative.

False positive rate = FP / (TN + FP)

Precision:

It tells us the correctness measure of predictions when the prediction is positive.

Precision = TP / (TP + FP)

Since all these metrics are derived from confusion matrix, it gives us a complete picture of the performance of the classifier.

Recall:

It is the same as sensitivity ie. the ability of the classifier to find all the positive samples.

F-1 measure:

Since we have two separate measures – precision and recall, we have a measurement that represents both of them. It uses harmonic mean as it punishes the extreme values more. Usually, f-measure is nearer to the smaller value of precision and recall. The formula is given by:

$$F\text{-}measure = \frac{2*Recall*Precision}{Recall + Precision}$$

Log loss:

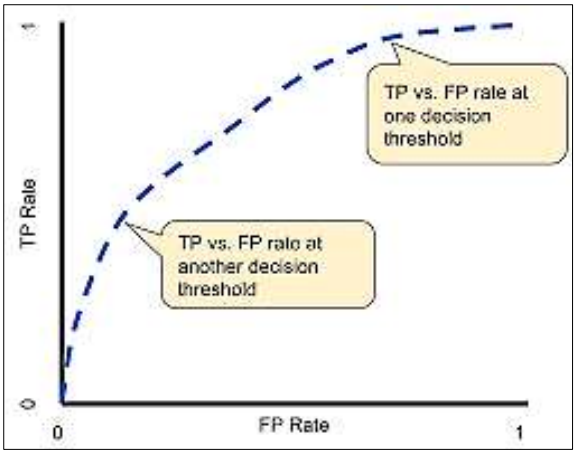
Log-loss is related to cross entropy and it measures the performance of a classification model where the prediction input is a probability between 0 and 1. The goal of the machine learning models is to minimize this value. The formula is given by:

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

Receiver Operating Characteristic curve (ROC):

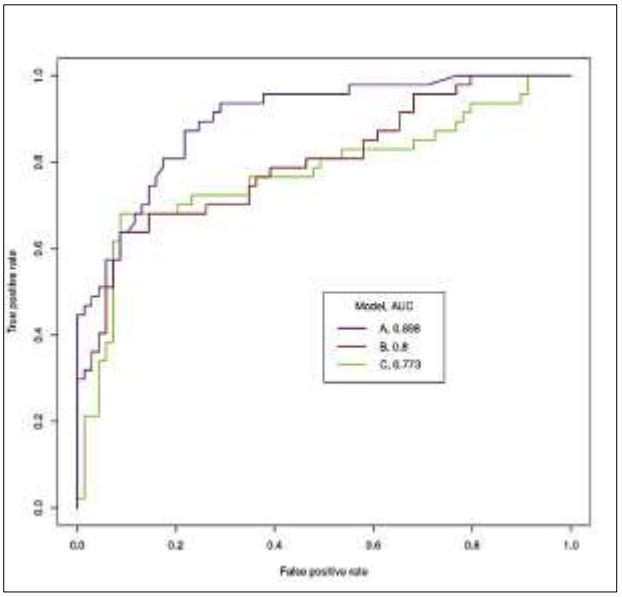
It illustrates the ability of a binary classifier as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

True positive rate (TPR), Recall, Sensitivity, probability of detection	False positive rate (FPR), Fall-out, probability of false alarm
$= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	$= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$



AUC:

It stands for Area under the ROC Curve. It measures the entire two-dimensional area underneath the entire ROC Curve [integration from (0,0) to (1,1). The closer an AUC model comes to 1, the better it is. The models with higher AUCs are preferred over those with lower AUCs.



IMPLEMENTATION: ADULT Dataset

The implementation was done on the adult dataset which is a **binary classification problem**. The observations have to be classified to state whether the person's income is >50K or <= 50K. First the data was loaded, pre-processed to replace the NaN values with the mode of that column. Also, the categorical data columns were modified to replace them with numerical data. Next, the dataset values were scaled using the StandardScaler. Scaling is very important so as to prevent one/more attributes from being given undue importance with respect to the others.

The classification model used was Logistic Regression. Further, the above described accuracy metrics were implemented on this Logistic Regression model.

a. **Classification accuracy**
The model gave a decent accuracy of 82.22%

b. **Null accuracy**
This is the accuracy obtained when the classifier predicts only the maximum occurring class (mode) for all observations. Thus, this is used to give an idea of the minimum required accuracy from our model. Null accuracy was 68.5%, which means that the Logistic Regression model is performing decently better than the most basic classifier.

c. **Confusion matrix**
As we know, the confusion matrix, gives values of True positives, True negatives, False positives and False negatives. From the model implementation we notice that- due to the imbalance of data (many more -1s than 1s in the final class), the number of false negatives is very high. Confusion matrix helps us derive many more accuracy measures. The have been described below.

d. **Classification error**
It is (1-accuracy) measure, and hence came out to be 17.77%

e. **Sensitivity:**
This value came out to be low due to the imbalance of data. Its value was 0.43. This measure needs to be maximized.

f. **Specificity:**
Since most of the class labels are -1, they also make up a large value for True Negatives. Specificity value came in at 0.94. **Although specificity, like sensitivity should be maximized, we see that there is a large imbalance, which must be corrected.**

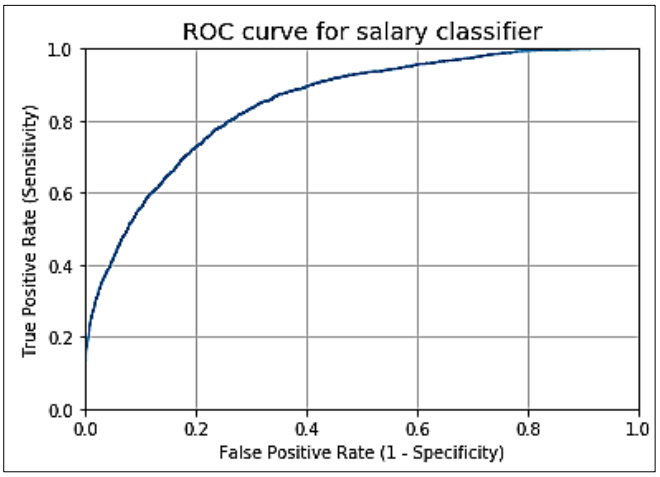
g. **False positive rate:**
It tells us the total wrongly classified positive labels, from the total observations labelled as positive. This value is also equal to (1-specificity) and was 0.05.

h. **Precision:**
This measure tells us the correctly classified positive labels. The value obtained was 0.70.

i. **F-score:**
It is the weighted average of the precision and recall values (as given in the formula above) and the value came in at 0.80.

j. **Log loss:**
The lower the log-loss, the better it is for the model. The log loss for the model was 6.13.

k. **ROC Curve:**
As explained above, the ROC uses false positive rate and true positive rates for the **different threshold values** set for the algorithm, and plots a graph. It gives us a good idea of the models performance.

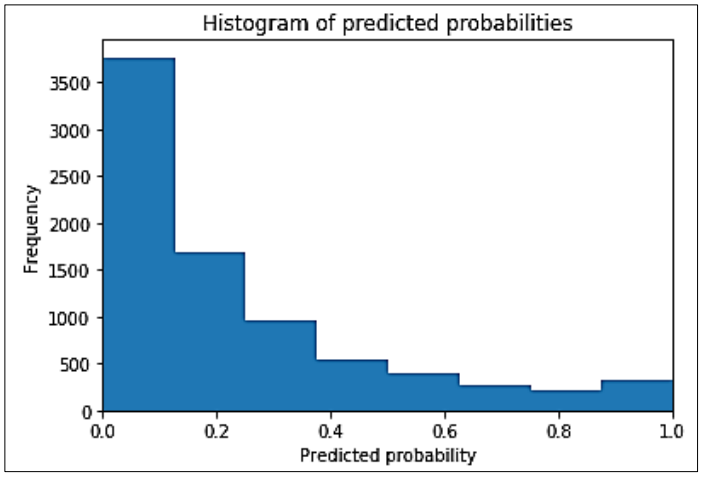


l. **AUC:**
The AUC represents the area under the ROC curve. The model with greater AUC is preferred over a model with lower AUC. The maximum possible AUC value for a model is 1. For our model, it came in at 0.85, which is good.

ADJUSTING CLASSIFICATION METHOD TO IMPROVE ACCURACY

Further, inferring from the code given in the tutorial, I also adjusted the Logistic Regression probability threshold to increase the number of True Positive classifications. There is **a trade-off between sensitivity and specificity.** Since in our model, the

difference was to large, the gap had to be bridged. From the below histogram, we see that most of the classifications for class = 1 occur at a probability < 0.3. Thus threshold must be decreased.



On reducing the classification threshold for class = 1 from 0.5 to 0.3, the **sensitivity** of the model increased from 0.43 to 0.655. On the other hand, the **specificity** decreased from 0.94 to 0.84. While both have to be maximized, the difference between the two should not be too large. Thus, after reducing the threshold probability for class = 1, the model would perform better.

Confusion matrix before modification:

```
[[5839  354]
 [1093  855]]
```

Confusion matrix after modification:

```
[[5235  958]
 [ 672 1276]]
```

RESULT:

Thus, the accuracy metrics for classification were dealt with in detail for the given dataset.

SCREENSHOT