به نام او، به یاد او، برای او

Machine Learning

Performance Metrics

Dr. Ali Valinejad

valinejad.ir
valinejad@umz.ac.ir

- **Performance Metrics for *Classification* Problems**
- **Performance Metrics for *Regression* Problems**

Performance Metrics for *Classification* Problems

Performance Metrics for *Classification* Problems

- After a classification model has been trained, a test data set is run against the model to determine performance.

Various *performance metrics* to evaluate *predictions*:

- Confusion Matrix
- Classification Accuracy
- Classification Report
  - Precision
  - Recall or Sensitivity
  - Specificity
  - Support
  - F1 Score
- LOGLOSS (Logarithmic Loss)
- AUC (Area Under ROC curve)

Performance Metrics for *Classification* Problems

❖ Confusion Matrix

It is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes.
A confusion matrix is nothing but a table with two dimensions: “Actual” and “Predicted”

Four Quadrant Measurement on “Performance” of a model.

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

Performance Metrics for *Classification* Problems

❖ Confusion Matrix

It is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes.
A confusion matrix is nothing but a table with two dimensions: “Actual” and “Predicted”

Four Quadrant Measurement on “Performance” of a model.

Number correctly predicted as the class

Number incorrectly predicted as not the class

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

Number incorrectly predicted as the class, when it is not that class.

Number correctly predicted not as the class

We can use confusion_matrix function of sklearn.metrics to compute Confusion Matrix of our classification model.

Performance Metrics for *Classification* Problems

❖ Classification Accuracy

It is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made. We can easily calculate it by confusion matrix.

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

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

We can use *accuracy_score* function of *sklearn.metrics* to compute accuracy of our classification model.

Performance Metrics for *Classification* Problems

❖ Classification Report

This report consists of the scores of *Precision*, *Recall*, *F1-score* and *Support*.

- *Precision*:

  Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model.

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

- *Recall* or *Sensitivity* or *True Positive Rate*(TPR) :

  Recall may be defined as the number of positives returned by our ML model.

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

- *Specificity*:

  Specificity, in contrast to recall, may be defined as the number of negatives returned by our ML model.

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

- *F1 Score*:

  This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0.

  $\boldsymbol{F1}$ score = $\mathbf{2} * (\boldsymbol{precision} * \boldsymbol{recall}) / (\boldsymbol{precision} + \boldsymbol{recall})$

  F1 score is having equal relative contribution of precision and recall.

- *Support*:

  Support may be defined as the number of samples of the true response that lies in each class of target values.

We can use *classification_report* function of *sklearn.metrics* to get the classification report of our classification model.

Performance Metrics for *Classification* Problems

❖ LOGLOSS (Logarithmic Loss)

It is also called Logistic regression loss or cross-entropy loss.
It basically defined on probability estimates and measures the performance of a classification model where the input is a probability value between 0 and 1.
It can be understood more clearly by differentiating it with accuracy. As we know that accuracy is the count of predictions (predicted value = actual value) in our model whereas Log Loss is the amount of uncertainty of our prediction based on how much it varies from the actual label.
With the help of Log Loss value, we can have more accurate view of the performance of our model.

We can use *log_loss* function of *sklearn.metrics* to compute Log Loss.

Performance Metrics for *Classification* Problems

❖ AUC (Area Under ROC curve)

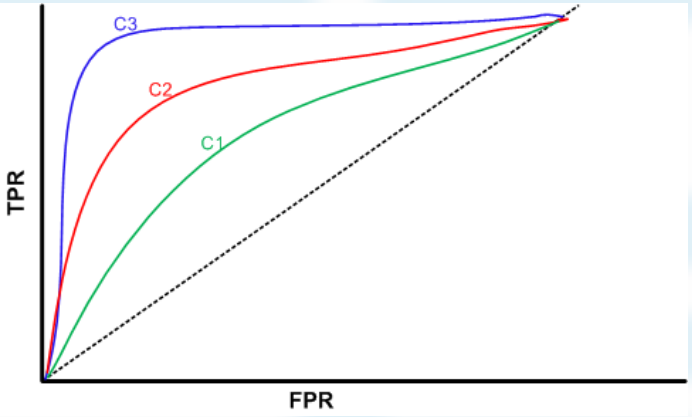
AUC (Area Under Curve)-ROC (Receiver Operating Characteristic) is a performance metric, based on varying threshold values, for classification problems. As name suggests, ROC is a probability curve and AUC measure the separability. In simple words, AUC-ROC metric will tell us about the capability of model in distinguishing the classes. Higher the AUC, better the model.

Mathematically, it can be created by plotting TPR (True Positive Rate) i.e. Sensitivity or recall vs FPR (False Positive Rate) i.e. 1-Specificity, at various threshold values.

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

$$1-Specificity=FPR = \frac{FP}{TN + FP}$$

Following is the graph showing ROC having TPR at y-axis and FPR at x-axis:



We can use roc_auc_score function of sklearn.metrics to compute AUC-ROC.

Performance Metrics for *Classification* Problems

Example:

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import log_loss

X_actual = [1, 1, 0, 1, 0, 0, 1, 0, 0, 0]
Y_predic = [1, 0, 1, 1, 1, 0, 1, 1, 0, 0]
results = confusion_matrix(X_actual, Y_predic)
print ('Confusion Matrix :')
print(results) print ('Accuracy Score is',accuracy_score(X_actual, Y_predic))
print ('Classification Report : ')
print (classification_report(X_actual, Y_predic)) print('AUC-
ROC:',roc_auc_score(X_actual, Y_predic))
print('LOGLOSS Value is',log_loss(X_actual, Y_predic))
```

Output:

```
Confusion Matrix :
[
  [3 3]
  [1 3]
]
Accuracy Score is 0.6
Classification Report :
              precision    recall  f1-score   support

           0       0.75      0.50      0.60         6
           1       0.50      0.75      0.60         4

   micro avg       0.60      0.60      0.60        10
   macro avg       0.62      0.62      0.60        10
weighted avg       0.65      0.60      0.60        10

AUC-ROC:  0.625
LOGLOSS Value is 13.815750437193334
```

Performance Metrics for *Regression* Problems

Performance Metrics for *Regression* Problems

- After a regression model has been trained, a test data set is run against the model to determine performance.
- Performance is measured as a *cost function* (or *loss function*) between the expected result, ($y^{(i)}$), and predicted result, ($h\left(x^{(i)}\right)$).

Various *performance metrics* to evaluate *predictions*:

- Mean Absolute Error (MAE)
- Mean Square Error (MSE)
- R Squared (R^2)

Performance Metrics for *Regression* Problems

❖ Mean Absolute Error (MAE)

It is the simplest error metric used in *regression* problems.
It is basically the sum of average of the absolute difference between the predicted and actual values.
In simple words, with MAE, we can get an idea of how wrong the predictions were.
MAE does not indicate the direction of the model i.e. no indication about underperformance or overperformance of the model.

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|h\left(x^{(i)}\right) - y^{(i)}\right|$$

$x^{(i)}$=Input Value

$y^{(i)}$=Actual Output Value

$h\left(x^{(i)}\right)$= Predicted Output Value.

Performance Metrics for *Regression* Problems

- **Mean Square Error (MSE)**

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$x^{(i)}$=Input Value

$y^{(i)}$=Actual Output Value

$h\left(x^{(i)}\right)$= Predicted Output Value.

Performance Metrics for *Regression* Problems

❖ R Squared (R^2)

R Squared metric is generally used for explanatory purpose and provides an indication of the goodness or fit of a set of predicted output values to the actual output values.

$$R\ Squared = 1 - \frac{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2}{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \bar{y})^2}$$

The numerator is MSE.
The denominator is the variance in $y^{(i)}$ values for $i = 1,2,\dots,m$.

$x^{(i)}$=Input Value,

$\bar{y}$=Mean of $y^{(i)}\ for\ i = 1,2,\dots,m$,

$y^{(i)}$=Actual Output Value,

$h\left(x^{(i)}\right)$= Predicted Output Value.

Performance Metrics for *Regression* Problems

Example:

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

X_actual = [5, -1, 2, 10]
Y_predic = [3.5, -0.9, 2, 9.9]
print ('R Squared =', r2_score(X_actual, Y_predic))
print ('MAE =', mean_absolute_error(X_actual, Y_predic))
print ('MSE =', mean_squared_error(X_actual, Y_predic))
```

Output:

```
R Squared = 0.9656060606060606
MAE = 0.42499999999999993
MSE = 0.5674999999999999
```

1. https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_algorithms_performance_metrics.htm
2. Data Analytics(CS40003)- Dr. Debasis Samanta, Associate Professor, Department of Computer Science & Engineering.