



CAS ETH Machine Learning in Finance and Insurance

BLOCK I. Introduction to Machine Learning.

Performance measures for classification problems

Dr. A. Ferrario, ETH Zurich and UZH

Repetita iuvant. The performance assessment of a machine learning model depends on the task at hand

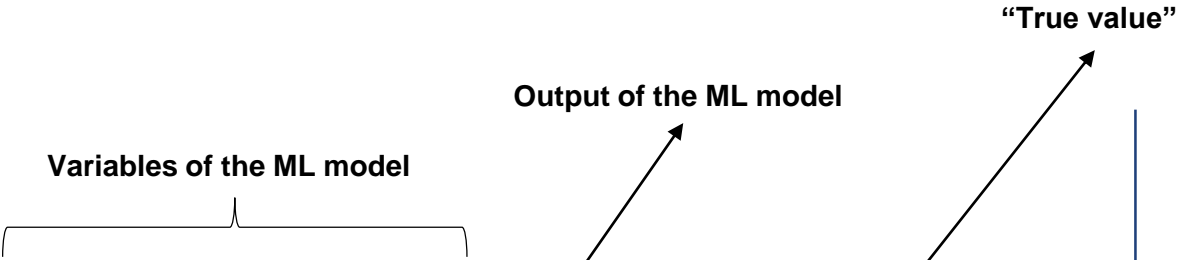
goal of these notes

	Characteristic of Y	Examples	Performance Measures
Classification	<ul style="list-style-type: none">A finite set of outcomes	<ul style="list-style-type: none">School gradesCredit yes/noChurn yes/noHealthy/Not healthy	<ul style="list-style-type: none">Q: “<i>How badly did I classify the outcomes?</i>”AccuracyPrecision and recallF1Others: Area Under Curve (AUC)
Regression	<ul style="list-style-type: none">A continuous variable	<ul style="list-style-type: none">Customer lifetime valueFinancial figures (e.g. revenue)Physical quantities: weight, length, time etc.	<ul style="list-style-type: none">Q: “<i>How badly did I approximate the numerical scores?</i>”R^2Mean Squared Error (MSE)

Let us consider a use case from insurance to introduce the problem of measuring the performance of machine learning models in (binary) classification problems

- Let “*Best Motor Insurance AG*” be a motor insurance operating in Switzerland. Despite its excellent customer service and competitive prices, churn of high value customers is an important source of risk for this company, which operates in a highly dynamic market with a lot of competitors.
- Opportunity: to predict customer churn in 1 year and target high-value + high likelihood to churn customers with an ad-hoc retention campaign
- Consider 100 customers in the “Ducati Monster” portfolio of the *Best Motor Insurance AG*. We know that 15 customers churned, 85 did not churn (ground truth/true classes).
- Suppose the data scientists of the *Best Motor Insurance AG* trained a machine learning model to predict customer churn (in one year). Its performance is tested on the 100 customers (test data). To evaluate its performance, we would like to compare the 100 model predictions (“churn” vs. “not churn”) to the ground truth/true classes of the 100 customers.

Overview of the performance evaluation on the 100 test customers



The diagram illustrates the components of the performance evaluation. A bracket labeled "Variables of the ML model" spans the columns "Variable 1", "...", and "Variable N". An arrow labeled "Output of the ML model" points to the "Prediction" column. Another arrow labeled "True value" points to the "Ground truth/True Class" column.

Customer	Variable 1	...	Variable N	Prediction	Ground truth/True Class
C1	0	0
C2				1	0
C3				1	1
C4				0	0
C5				1	1
...	0	1
C100				1	1

Is there a way to compare the predictions to the ground truth/true classes and measure the model performance?

Our encoding: 0 = not churned. 1= churned.

We compare the (binary) prediction to the ground truth/true class for each customer and collect results in the confusion matrix

Customer	Variables of the ML model			Prediction	Ground truth/True Class
	Variable 1	...	Variable N		
C1	0	0
C2				1	0
C3				1	1
C4				0	0
C5				1	1
...	0	1
C100				1	1

Our encoding: 0 = not churned. 1= churned.

Confusion matrix/misclassification table

		TRUE CLASS	
		1	0
PREDICTED CLASS	1	True Positives (TP)	False Positives (FP)
	0	False Negatives (FN)	True Negatives (TN)

1="churned"

0="not churned"

1= predicted "churn"

0=predicted "not churn"

The confusion matrix collects all relevant information to assess the performance of the model in the (binary) classification problem

- The **confusion matrix/misclassification table** collects all possible cases of (mis)classification
- **True positives** = #customer who churned that the model correctly classified (prediction is “churn”)
- **True negatives** = #customers who did not churn that the model correctly classified (prediction is “not churn”)
- **False positives** = #customers who did not churn that the model incorrectly classified (prediction is “churn”)
- **False negatives** = #customers who did churn that the model correctly classified (prediction is “not churn”)

Confusion matrix/misclassification table

		TRUE CLASS	
		1	0
PREDICTED CLASS	1	True Positives (TP)	False Positives (FP)
	0	False Negatives (FN)	True Negatives (TN)

1=“churned”

0=“not churned”

1= predicted “churn”

0=predicted “not churn”

Let us discuss the elements of the confusion matrix

- By definition:
 - $TP + FP + FN + TN = 100$
 - $TP + FN = \text{\#customer who churned}$
 - $TN + FP = \text{\#customers who did not churn}$
- The ML model correctly classified 10 out of 15 customers who churned
- The ML model correctly classified 60 out of 85 customers who did not churn

Confusion matrix/misclassification table

		TRUE CLASS	
		1	0
PREDICTED CLASS	1	10 (TP)	25 (FP)
	0	5 (FN)	60 (TN)
		=15 customers who churned	=85 customers who did not churn

Let us start with the Case Study

Modeling – Assessing Model (Performance). Classification.

- **Accuracy:** $(TP + TN) / (TP + TN + FP + FN)$

→ proportion of correctly classified data points

- **Precision:** $TP / (TP + FP)$

→ proportion of true positives among all data points classified as positives

- **Sensitivity/Recall/True Positive Rate:** $TP / (TP + FN)$

→ proportion of true positives correctly classified

- **Specificity:** $TN / (TN + FP)$

proportion of true negatives correctly classified

- **Note:** False Positive Rate=1-Specificity

- **F1 Score:** $2TP / (2TP + FP + FN)$

→ harmonic average of precision and recall

Confusion matrix/misclassification table

		TRUE CLASS	
		1	0
PREDICTED CLASS	1	10 (TP)	25 (FP)
	0	5 (FN)	60 (TN)

Accuracy=(10+60)/100=70.0%

Precision=10/(10+25)=28.5%

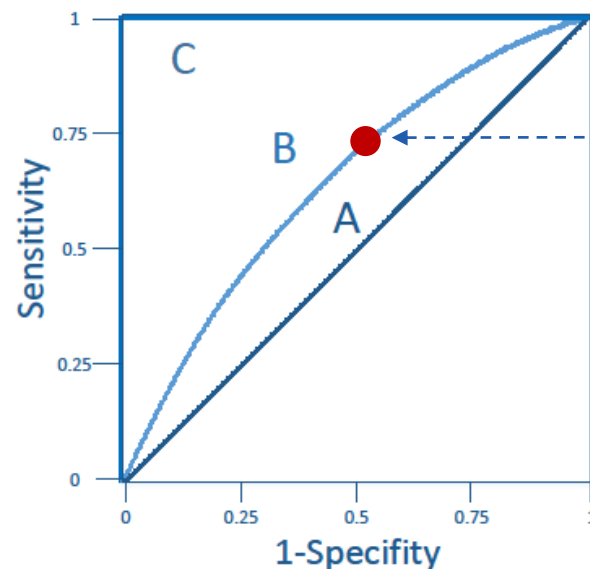
Sensitivity/Recall/True Positive Rate=10/(10+5)=66.7%

F1=40/(40+25+5)=57.1%

Specificity=60/(60+25)=70.6%

AUROC, ROC and AUC

- **Receiver Operating Characteristic: ROC**
- **Area Under ROC = AUROC**, also denoted by AUC^* , or Area Under Curve
- T. Fawcett, 2006. An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874, available at <https://people.inf.elte.hu/kiss/11dwhdm/roc.pdf>



1. Each point on the ROC curve corresponds to a choice of classification threshold

1. For each choice of threshold, the coordinates

1-Specificity = $1 - TN / (TN + FP)$ = **False Positive Rate**

Sensitivity = $TP / (TP + FN)$ = **True Positive Rate**

are computed and the corresponding point plotted.

[*https://en.wikipedia.org/wiki/Receiver_operating_characteristic](https://en.wikipedia.org/wiki/Receiver_operating_characteristic)

AUROC, ROC and AUC in Python: pseudo-code

```
# let model be a trained machine learning model
# compute the probabilities of churn on test data - select the correct array column! .predict_proba() returns two probabilities
y_scores = model.predict_proba(X_test)[:, 1]

# Compute ROC curve and ROC area
fpr, tpr, _ = roc_curve(y_test, y_scores)
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc="lower right")
plt.show()
```