Text Classification (Part I)

[DAT640] Information Retrieval and Text Mining

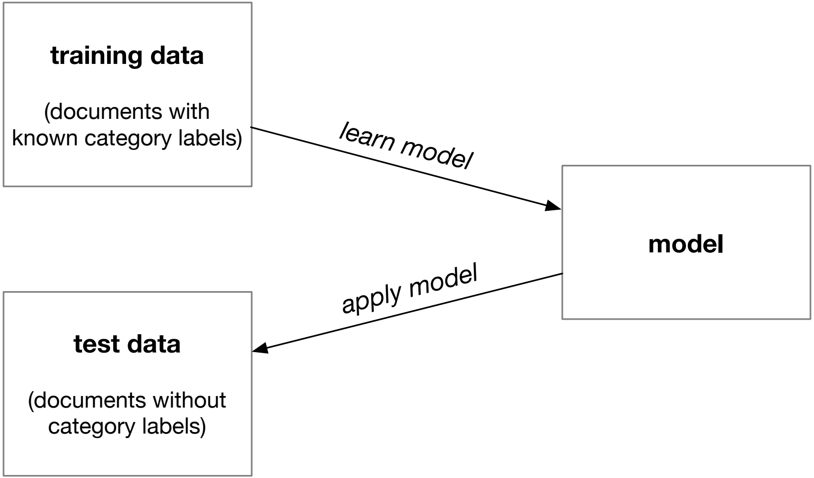
Krisztian Balog
University of Stavanger

August 20, 2019

Text classification

- **Classification** is the problem of assigning objects to one of several predefined categories
  - One of the fundamental problems in machine learning, where it is performed the basis of a training dataset (instances whose category membership is known)
- In **text classification** (or **text categorization**) the objects are text documents
- Binary classification (two classes, $0/1$ or -$/+$)
  - E.g., deciding whether an email is spam or not
- Multiclass classification (n classes)
  - E.g., Categorizing news stories into topics (finance, weather, politics, sports, etc.)

General approach



Formally

- Given a training sample $(\boldsymbol{X}, y)$, where $\boldsymbol{X}$ is a set of documents with corresponding labels y, from a set $\boldsymbol{Y}$ of possible labels, the task is to learn a function $f(\cdot)$ that can predict the class $y' = f(x)$ for an unseen document x.

Evaluation

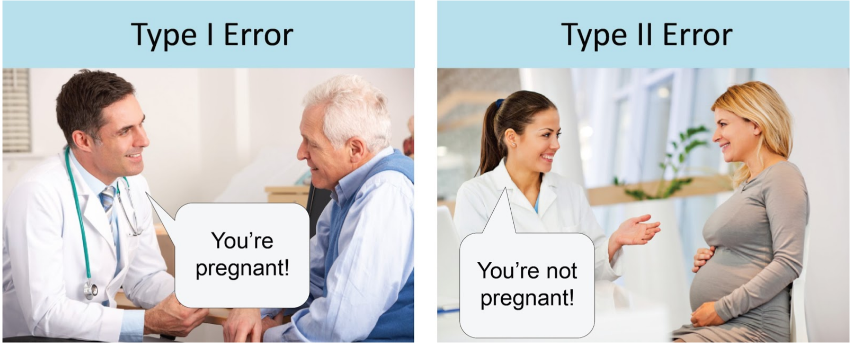
- Measuring the performance of a classifier
  - Comparing the predicted label y' against the true label y for each document in some set dataset
- Based on the number of records correctly and incorrectly predicted by the model
- Counts are tabulated in a table called the **confusion matrix**
- Compute various **performance measures** based on this matrix

Confusion matrix

		Predicted class	
		negative	positive
Actual class	negative	true negatives (TN)	false positives (FP)
	positive	false negatives (FN)	true positives (TP)

- False positives = Type I error (“raising a false alarm”)
- False negatives = Type II error (“failing to raise an alarm”)

Type I vs. Type II errors[1]



[1]Source:
https://www.analyticsindiamag.com/understanding-type-i-and-type-ii-errors/

Evaluation measures

- Summarizing performance in a single number
- **Accuracy**
  Number of correctly classified items out of all items

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

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

- **Error rate**
  Number of incorrectly classified items out of all items

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

Evaluation measures (2)

- **Precision**
  Number of items correctly identified as positive out of the total items identified as positive

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

- **Recall** (also called Sensitivity or True Positive Rate)
  Number of items correctly identified as positive out of the total actual positives

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

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Evaluation measures (3)

- **F1-score**
  The harmonic mean of precision and recall

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

		predicted	
		-	$+$
actual	-	TN	FP
	$+$	FN	TP

Evaluation measures (4)

- **False Positive Rate (Type I Error)**
  Number of items wrongly identified as positive out of the total actual negatives

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

- **False Negative Rate (Type II Error)**
  Number of items wrongly identified as negative out of the total actual positives

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

		predicted	
		-	+
actual	-	TN	FP
	+	FN	TP

Exercise #1 (paper-based)

Compute Accuracy, Precision, Recall, F1-score, False Positive Rate, and False Negative Rate for a classifier that made the following predictions

Id	Actual	Predicted
1	+	-
2	+	+
3	-	-
4	+	+
5	+	-
6	+	+
7	-	-
8	-	+
9	+	-
10	+	-

Exercise #2

Implement the computation of Accuracy, Precision, Recall, and F1-score in Python.

- Complete the notebook: `exercises/lecture_02/exercise_2.ipynb`

Discussion

Question

Which of the Type I/II errors would be more severe for a spam classifier?
Which of these measures would be most appropriate for evaluating a spam classifier?

Model development

- In practice, we don't have access to the actual category labels
- **How can we evaluate the performance of the model during development?**

Model development

- In practice, we don't have access to the actual category labels
- **How can we evaluate the performance of the model during development?**
- **Idea: hold out part of the training data for testing**

Two strategies

- **Single train/validation split**
  - Split the training data into $X\%$ training split and $100 - X\%$ validation split (an 80/20 split is common)
- **k-fold cross-validation**
  - Partition the training data randomly into k folds
  - Use $k-1$ folds for training and test on the kth fold; repeat k times (each fold is used for testing exactly once)
  - k is typically 5 or 10
  - Extreme: k is the number of data points, to maximize the number of training material available (called "leave-one-out" evaluation)

Assignment 1A