university of groningen

Quantitative Text Analysis – Essex Summer School

Supervised machine learning

dr. Martijn Schoonvelde

University of Groningen

Today's class

- Supervised text analysis
- Lab session
- Flash talks: Tanja and Federico

A definition of supervised machine learning

"A form of machine learning where we aim to predict a variable, that, for at least part of our data, is known" (Van Atteveldt, 2022: p. 114)

- We estimate a model based on some complete data, and then use the model to predict the expected outcome for some new cases, for which we do not know the outcome yet

In the context of QTA (supervised text analysis), we rely on textual data to make our predictions

Supervised text analysis

- Three broad types of analysis (Boumans & Trilling 2016), from most deductive to most inductive:
  - **counting and dictionary methods**: the researcher can fully specify relevant features, and will categorise text accordingly
  - **supervised methods**: the researcher knows how to categorise documents, and uses machine learning methods to learn which features drive this categorisation
  - **unsupervised methods**: the researcher uses qta tools to learn about textual categorisation inductively

Vector space models versus language models

Two general ways of thinking about representations of text

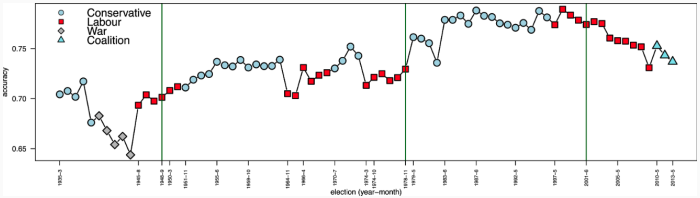
- **Vector space model** – feature matrix as an input towards answering a research question. Agnostic as to how it was generated.
- **Language models** – feature matrix as the outcome of an underlying language process. Goal is to learn about that underlying process

Much overlap and often you can do the same thing using either using either approach (Grimmer, Roberts & Stewart, 2022)

In general these approaches connect to our research approach: supervised, semi-supervised and unsupervised methods

Examples of text-based supervised machine learning

- Detecting Islamophobic hate speech on social media (Vidgen & Yasseri, 2020)
- Temporal focus of campaign communication (Müller, 2021)
- Measuring polarization in UK House of Commons (Peterson & Spirling, 2018)



Source: Peterson & Spirling, 2018

Workflow and terminology

1. Categorize a set of documents by hand – create a labeled dataset
2. Divide categorized documents in a training set and a test set
3. Use a supervised algorithm to 'learn' the relationship between the known labels and the features in the training set
4. Evaluate the classifier based on whether it correctly categorizes documents in the test set
5. Use the classifier to categorize unseen texts (not part of the training and the test set)

machine learning lingo	statistics lingo
feature	independent variable
label	dependent variable
labeled dataset	dataset with both independent and de
to train a model	to estimate
classifier (classification)	model to predict nominal outcomes
to annotate	to (manually) code (content analysis)

Source: Van Atteveldt *et al.* 2022

Supervised machine learning and text analysis

In the 1960s, two statisticians, Frederick Mosteller and David Wallace used a statistical approach to identify the authors of 12 disputed papers in the *Federalist Papers*

- Alexander Hamilton, James Madison, and John Jay
- Concluded that in fact it was Hamilton who authored these papers

Early application of supervised machine learning to engage in authorship attribution or stylometry

- Identification of an author of text based on the semantic content and linguistic style of their writing

Evaluating a classifier: accuracy

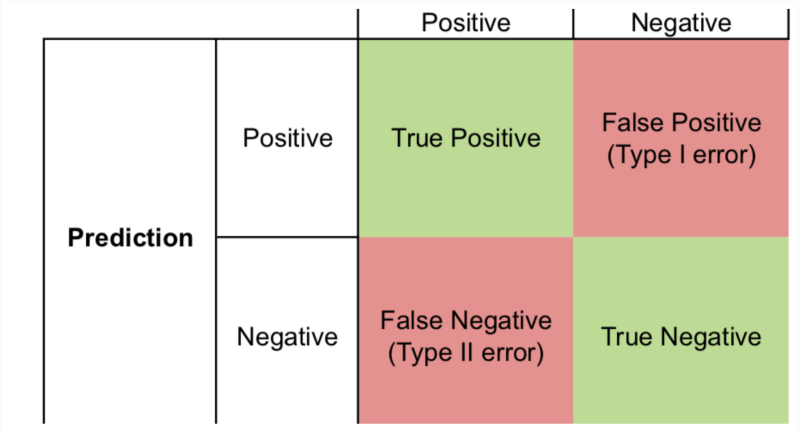
Let's say we have 100 Federalist papers for which we know the author (Madison or either Jay and Hamilton ('other').

- 12 are written by Madison and 88 by other

It would be very easy to accurately predict authorship by guessing other each time. We would be correct 88% of the time. However, this is also quite meaningless. We would not identify any of the Madison papers.

Evaluating a classifier: confusion matrix

A confusion matrix is a table that summarizes the performance of a classifier.



Evaluating a classifier: confusion matrix

- Let's say we build a classifier and are able to predict 10 out of 12 Madison papers. But we also assign 3 papers incorrectly to Madison

		Reality	
		Madison	Other
Prediction	Madison	10	3
	Other	2	85

Evaluating a classifier: confusion matrix

		Reality	
		Madison	Other
Prediction	Madison	10 (= **TP**)	3 (= **FP**)
	Other	2 (= **FN**)	85 (= **TN**)

Evaluating a supervised classifier: precision and recall

Precision, recall and the F1 score are frequently used to assess classification performance:

- Recall: **TP / (TP + FN)**
  - *Do we identify all papers by Madison?*
- Precision: **TP / (TP + FP)**
  - *Do we identify only papers by Madison?*
- F1 score is a harmonic mean of precision and recall **2 * (Precision * Recall) / (Precision + Recall)**.

Evaluating supervised methods: precision and recall

- Recall: **TP / (TP + FN)**
  - $\rightarrow$ 10 / (10 + 2) = 0.83
- Precision: **TP / (TP + FP)**
  - $\rightarrow$ 10 / (10 + 3) = 0.77
- F1 score is a harmonic mean of precision and recall **2 * (Precision * Recall) / (Precision + Recall)**.
  - $\rightarrow$ 2*(0.77*0.83) / (0.77 + 0.83) = 0.80

How to find the 'best' model?

Precision, recall and F1 scores are useful, but it's up to us which classifier to choose. And as always, our decision will depends on our research goals.

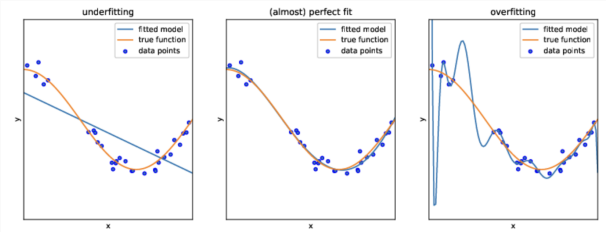
- If we are really determined not to miss any Madison papers, we may prioritise recall
- If we want to be certain that the papers we identify are indeed by Madison we may prioritise precision

F1 scores compromise between both but we don't want precision and recall to diverge too much

- For classifiers that also give us a prediction probabilities, we can also display a ROC curve, a plot that displays true positive against false positive at different probability thresholds

Generalisation and overfitting

- **Generalisation**: a classifier learns to correctly predict labels from given inputs not only in previously seen samples but also in previously unseen samples
- **Overfitting**: a classifier learns to correctly predict labels from given inputs in previously seen samples but fails to to do in in previously unseen samples

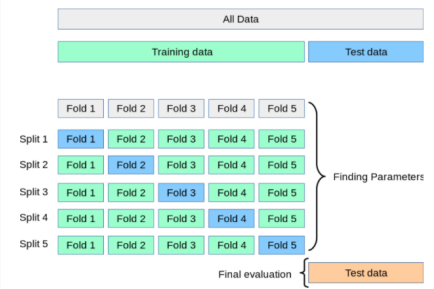


Source: Van Atteveldt, Trilling, Arcila Calderón, 2022

Cross validation

If we test the performance of multiple classifiers on our test data, we run the risk of overfitting our model to the test data

- One approach to address this issue is called k-fold cross validation where we separate our data into k separate training and test sets, and average our model results.
- If our classifier generalizes well, we should expect that our metric of choice is similar in all folds



Source `https://stats.stackexchange.com/`

Naïve Bayes classifier

- Use Bayes' rule to predict that a document has a certain label (e.g., positive or negative)
  - Also can be applied when we have more two labels (e.g., documents about sports, the economy or politics)
- Naïve assumption – all features are independent from each other, as if they are random draws from a vocabulary

Bayes' theorem:

$$P(label \mid features) = \frac{P(features \mid label)\, P(label)}{P(features)}$$

`NB: thanks to Stefan Müller for letting me borrow the next few slides`

Naïve Bayes classifier

Intuition: If we observe the term "fantastic" in a text, how likely is this text a positive review?

1. Determine frequency of positive and negative reviews (prior).
2. Assess probability of features given a particular class.
3. Get probability of a document belonging to each class (posterior).
4. Which posterior is highest?

Naïve Bayes classifier

Advantages

- Simple, fast, effective
- Relatively small training set required (if classes no too imbalanced). Easy to obtain probabilities

Disadvantages

- Assumption of conditional independence is problematic
  - In a document about football we expect words like ball, victory, league etc to cluster
- If feature is not in training set, it is disregarded for the classification
  - But we can include a smoothing parameter

Naïve Bayes in quanteda

```
library(quanteda); library(quanteda.textmodels)
#get training set
dfmat_train <- tokens(c("positive bad negative horrible", "great fantastic nice")) %>% dfm()
class <- c("neg", "pos")

#train model
tmod_nb <- textmodel_nb(dfmat_train, class)

#get unlabelled test set
dfmat_test <- tokens(c("bad horrible afwul", "nice bad great", "great fantastic")) %>% dfm()

#predict class
predict(tmod_nb, dfmat_test, force = TRUE, type = "probability")

            neg       pos
text1 0.8573572 0.1426428
text2 0.2730748 0.7269252
text3 0.1712329 0.8287671
```

Other supervised classifiers

- Support Vector Machine, Logistic regression and other classifiers implemented in **quanteda.textmodels**
- Check out the book "Supervised Machine Learning for Text Analysis in R" by Emil Hvitfeldt and Julia Silge: `https://smltar.com/`
- Many, many tutorials online

Concluding thoughts

- Categorization depends on the quality of labeled data
  - Classifier is trained to mimic a 'gold standard' – but what if that gold standard is not so go golden after all?
- Error analysis – are our mispredictions systematic?