Computational Communication Science 2 Week 7 - Lecture »Rule-based vs. Automated Text Classification«

Marthe Möller
Anne Kroon

a.m.moller@uva.nl, @marthemoller
a.c.kroon@uva.nl, @annekroon

May, 2022

Digital Society Minor, University of Amsterdam

Today

Rule-based Text Classification

Text Classification

Text classification: To assign a label to a text.

For example, to distinguish between:

- newspaper articles about sports vs. economics.
- reliable vs. unreliable information about vaccination.
- webpages about holding companies vs. financing companies.
- positive vs. negative movie reviews.

Studying Flaming (Example)

RQ: How problematic is flaming on Twitter?

Bag-of-words approach:

1. Create a list with all the swearwords that exist.
2. For each tweet in the dataset, use the list to count the number of swearwords
3. If a tweet contains X number of swearwords label it as flaming

Sentiment Analysis

We can add nuance by creating more rules.

For example, in sentiment analyses, we can include a rule telling the machine what to do in case of negation or modifiers.

"This movie is really not good."
"This movie is really good."

Rule-based Text Classifcation

Advantages of rule-based text classification:

- Simple and therefore transparent
- Cheap

Challenges of rule-based text classification:

- Not a suitable way to anayze latent or abstract variables
- You must know all the categories beforehand
- You must know and be able to express all the rules

From Rule-based to Automated

When it is easy for humans to decide to what class a text belongs, but we struggle to translate our decision process into straight-forward rules, we are likely to be better of using a form of automated text classification: Supervised Machine Learning.

SML

What is SML?

Yu, J., Ma, X., & Han, T. (2016). Four-Dimensional Usability Investigation of Image CAPTCHA. *arXiv preprint arXiv:1612.01067*.

What is SML?

Read more about this project in: Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv:1312.6229 [cs]*. Retrieved December 23, 2021, from http://arxiv.org/abs/1312.6229

What is ML?

Machine Learning: “a type of artificial intelligence in which computers use huge amounts of data to learn how to do tasks rather than being programmed to do them.”

Oxford Dictionary

What is SML?

Supervised Machine Learning (SML): "A form of machine learning, where we aim to predict a variable that, for a least part of our data is known."

Van Atteveldt, W., Trilling, D., & Calderon, C. A. (2022). *Computational analysis of communication*.

Wiley-Blackwell

What is SML?

"The goal of Supervised Machine Learning: estimate a model based on some data, and then use the model to predict the expected outcome for some new cases, for which we do not know the outcome yet."

Van Atteveldt, W., Trilling, D., & Calderon, C. A. (2022). *Computational analysis of communication.* Wiley-Blackwell

What is SML?

Machine Learning has a lot of similarities to regression analysis!

The principles behind SML

The principles behind SML

$y = constant + b_1 * x_1 + b_2 * x_2$

x_1 = bark? (0= no, 1 = yes)
x_2 = tail? (0 = no, 1 = yes)
y = Is this a dog? (0 = definitely no, 1 = definitely yes)

The principles behind SML

$$y = constant + b_1 * x_1 + b_2 * x_2$$

$$y = 0 + 0.8 * x_1 + 0.2 * x_2$$

The principles behind SML

$$y = 0 + 08 * 1 + 0.2 * 0$$

$$0.8 = 0 + 0.8 * 1 + 0.2 * 0$$

The principles behind SML

$0.8 = 0 + 0.8 * 1 + 0.2 * 0$

Classification: a predictive modeling problem where a class label is predicted for a given example of input data.

The principles behind SML

Machine Learning Lingo	Statistics Lingo
Feature	Independent variable
Label	Dependent variable
Labeled dataset	Dataset with both independent and dependent variables
To train a model	To estimate
Classifier	Model to predict nominal outcomes
To annotate	To (manually) code

Adapted from: Van Atteveldt, Trilling, & Arcilla (2021)

The principles behind SML

Machine Learning: using a (regression) formula to predict a label.

Traditional usage of formulas in CS: to explain

Usage of formulas in ML: to predict

Zooming out

We talked about:

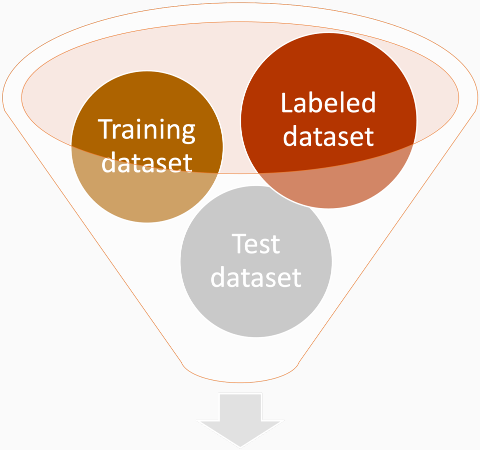
- Rule-based Text Classification
- Automated Text Classification: SML
- The principles behind SML

Next, we will talk about:

- Some commonly used SML models

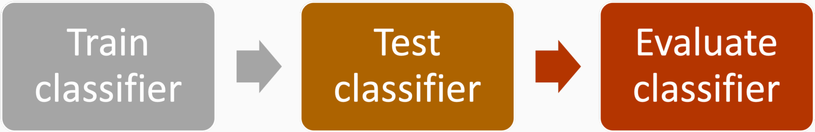
SML models

SML step by step

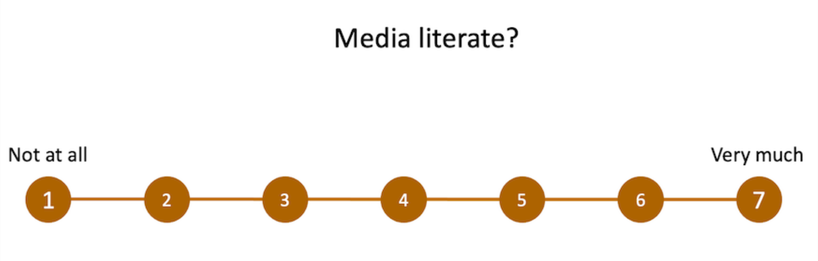


Machine Learning Process

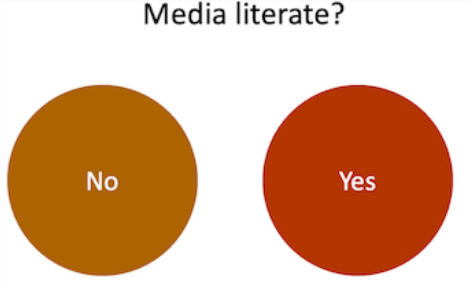
SML step by step



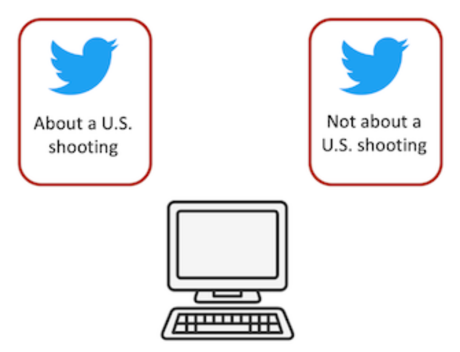
Regression



Logistic Regression

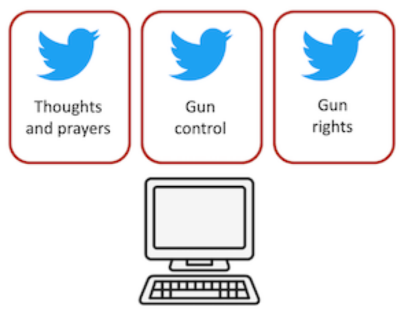


Logistic Regression



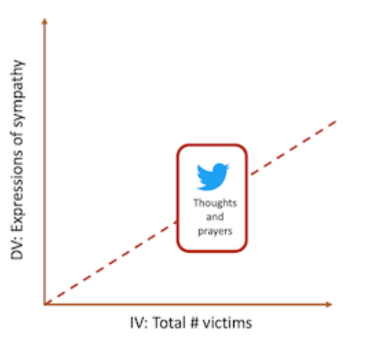
Zhang, Y., Shah, D., Foley, J., Abhishek, A., Lukito, J., Suk, J., Kim, S. J., Sun, Z., Pevehouse, J., & Garlough, C. (2019). Whose lives matter? mass shootings and social media discourses of sympathy and policy, 2012–2014. *Journal of Computer-Mediated Communication*, *24*(4), 182–202. https://doi.org/10.1093/jcmc/zmz009

Logistic Regression



Zhang, Y., Shah, D., Foley, J., Abhishek, A., Lukito, J., Suk, J., Kim, S. J., Sun, Z., Pevehouse, J., & Garlough, C. (2019). Whose lives matter? mass shootings and social media discourses of sympathy and policy, 2012–2014. *Journal of Computer-Mediated Communication*, *24*(4), 182–202. https://doi.org/10.1093/jcmc/zmz009

Logistic Regression



Zhang, Y., Shah, D., Foley, J., Abhishek, A., Lukito, J., Suk, J., Kim, S. J., Sun, Z., Pevehouse, J., & Garlough, C. (2019). Whose lives matter? mass shootings and social media discourses of sympathy and policy, 2012–2014. *Journal of Computer-Mediated Communication*, *24*(4), 182–202. https://doi.org/10.1093/jcmc/zmz009

What does this look like in code?

First, we need to read in the ingredients we need for SML.

```
import csv
from sklearn.model_selection import train_test_split

tweets = []
labels = []

with open(file) as fi:
    data = csv.reader(fi, delimiter='\t')
    for row in data:
        tweets.append(row[0])
        labels.append(row[1])

tweets_train, tweets_test, y_train, y_test = train_test_split(tweets,
     labels, test_size=0.2, random_state=42)
```

Where file is some file containg tweets (column 0) and their labels (column 1).

What does this look like in code?

Second, vectorize the texts that need to be labeled:

```
from sklearn.feature_extraction.text import (TfidfVectorizer)

tfidfvectorizer = TfidfVectorizer(stop_words="english")
X_train = tfidfvectorizer.fit_transform(tweets_train)
X_test = tfidfvectorizer.transform(tweets_test)
```

Where tweets_train and tweets_test are two lists with tweets (strings)

What does this look like in code?

Next, I train my machine and test it:

```
from sklearn.linear_model import (LogisticRegression)

logres = LogisticRegression()
logres.fit(X_train, labels_train)
y_pred = logres.predict(X_test)
```

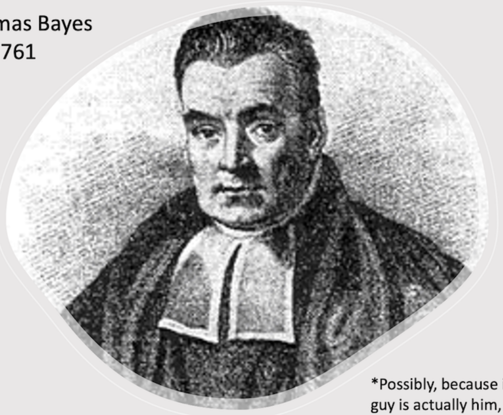
What does this look like in code?

To train a model based on a tf-idf vectorizer and Log Regression:

```
from sklearn.feature_extraction.text import (TfidfVectorizer)
from sklearn.linear_model import (LogisticRegression)

tfidfvectorizer = TfidfVectorizer(stop_words="english")
X_train = tfidfvectorizer.fit_transform(tweets_train)
X_test = tfidfvectorizer.transform(tweets_test)

logres = LogisticRegression()
logres.fit(X_train, labels_train)
y_pred = logres.predict(X_test)
```

Naïve Bayes

Possibly* Thomas Bayes
1702 – 1761



*Possibly, because it is unclear if this guy is actually him, but there is no other (claimed) portrait of him.

Naïve Bayes

$P(\mathrm{A} \mid \mathrm{B}) = \frac{\mathrm{P(B|A) \cdot P(A)}}{\mathrm{P(B)}}$

Mathematicians' language for: the probability of A if B is the case/present/true.

$P(\mathrm{label} \mid \mathrm{features}) = \frac{\mathrm{P(features|label) \cdot P(label)}}{\mathrm{P(features)}}$

What does this look like in code?

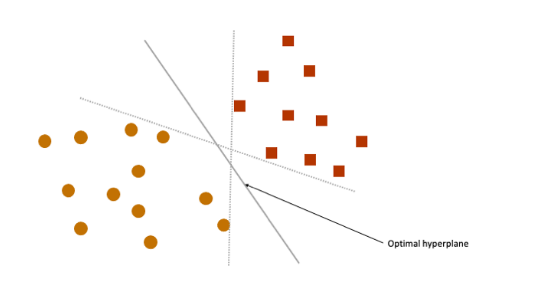
Let’s also train a model based on a count vectorizer and Naïve Bayes:

```
from sklearn.feature_extraction.text import (CountVectorizer)
from sklearn.naive_bayes import MultinomialNBB

countvectorizer = CountVectorizer(stop_words="english")
X_train = countvectorizer.fit_transform(texts_train)
X_test = countvectorizer.transform(texts_test)

nb = MultinomialNB()
nb.fit(X_train, labels_train)
y_pred = nb.predict(X_test)
```

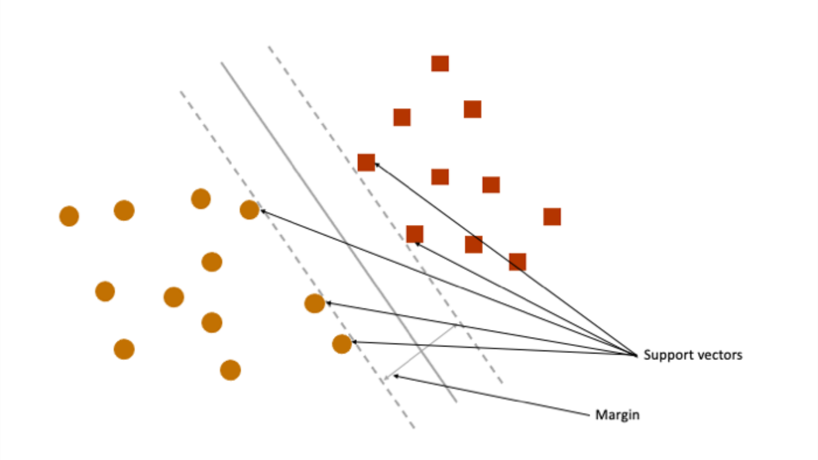
Support Vector Machines

SVMs aim to find a hyperplane in an N-dimensional pace that distinctly classifies the datapoints.

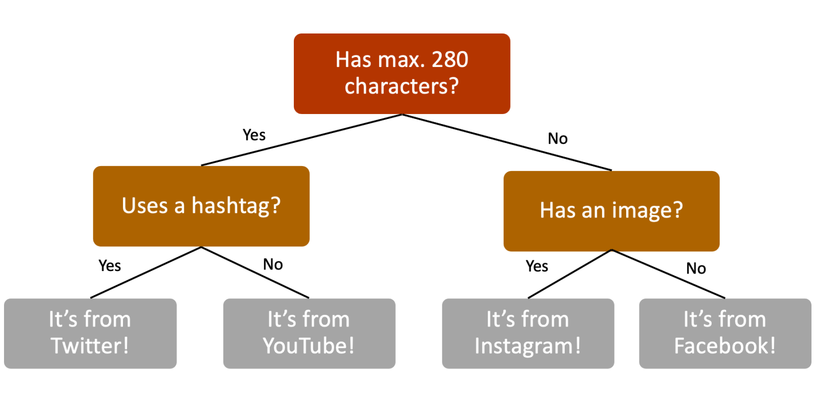
The best hyperplane is the one that has the maximum margin (distance) between the datapoints of both classes.



Support Vector Machines



Decision Trees and Random Forests



Decision Trees and Random Forests

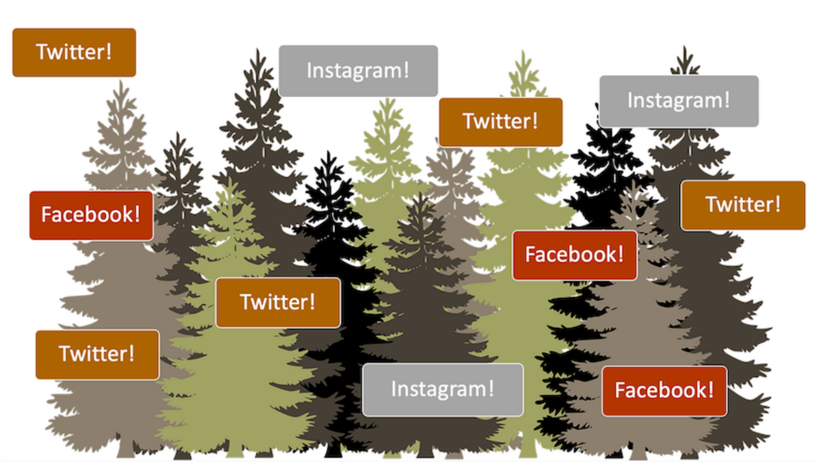
Advantages of decision trees:

- Transparency
- Suitable for non-linear relationships

Disadvatanges of decision trees:

- Loss of nuance due to yes/no-design
- Cannot correct early mistakes
- Prone to overfitting

Decision Trees and Random Forests



Recap

Many different models available for machine learning.

How do you know what is the best for your case? Try it out and validate!

Zooming out

We talked about:

- Rule-based Text Classification
- Automated Text Classification: SML
- The principles behind SML
- Some commonly used ML models

Next, we will talk about:

- Validating models

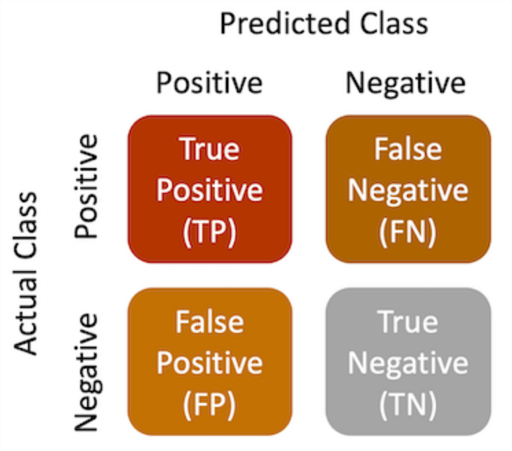
Validating models

Precision and Recall

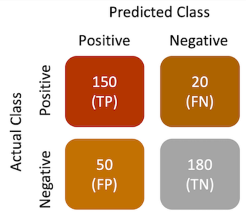
Precision quantifies the number of positive class predictions that actually belong to the positive cases.
OR: How much of what we found is actually correct?

Recall quantifies the number of positive class prediction made out of all positive examples in the dataset.
OR: How many of the cases that we wanted to find did we actually find?

Precision and Recall



Precision and Recall



Precision is calculated as: $\frac{\text{TP}}{\text{TP+FP}}$

In our case $\frac{150}{150+50}$ which is 0.75

Recall is calculated as $\frac{\text{TP}}{\text{TP+FN}}$

In our case $\frac{150}{150+20}$ which is 0.88

What does this look like in code?

A model based on a count vectorizer and Naïve Bayes:

```
from sklearn.feature_extraction.text import (CountVectorizer)
from sklearn.naive_bayes import MultinomialNBB

countvectorizer = CountVectorizer(stop_words="english")
X_train = countvectorizer.fit_transform(texts_train)
X_test = countvectorizer.transform(texts_test)

nb = MultinomialNB()
nb.fit(X_train, labels_train)
y_pred = nb.predict(X_test)
```

What does this look like in code?

Let's ask for a confusion matrix:

```
from sklearn.metrics import confusion_matrix

y_test = [0, 1, 1, 1, 0]
y_pred = [0, 0, 1, 1, 1]

print(confusion_matrix(y_test, y_pred))
```

```
[[1  1 ]
[ 1  2]]
```

What does this look like in code?

Let's get some metrics for validation:

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
            precision   recall f1-score  support

0       0.50     0.50     0.50       2
1       0.67     0.67     0.67       3

accuracy                       0.60       5
macro avg     0.58     0.58     0.58       5
weighted avg     0.60     0.60     0.60       5
```

F_1-score

F_1-score: The harmonic mean of precision and recall.

$F_1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

Precision and Recall

Table 2
Relationship classification performance and number of training tweets, random sampling approach.

		100	200	500	1000	2000	3000	4000
Linear support vector machine classifier	AC	0.63	0.65	0.70	0.73	0.80	0.84	0.91
	PC	0.45	0.48	0.59	0.62	0.76	0.80	0.90
	RC	0.38	0.43	0.51	0.59	0.71	0.79	0.86
	AUC	0.41	0.45	0.59	0.61	0.69	0.76	0.85
	KA	0.09	0.10	0.39	0.41	0.54	0.65	0.79
Naïve Bayes classifier	AC	0.63	0.65	0.71	0.75	0.82	0.86	0.91
	PC	0.42	0.46	0.62	0.68	0.81	0.86	0.92
	RC	0.27	0.33	0.47	0.49	0.61	0.69	0.79
	AUC	0.33	0.38	0.60	0.62	0.69	0.77	0.84
	KA	0.08	0.13	0.39	0.40	0.56	0.67	0.78
Logistic regression classifier	AC	0.66	0.67	0.71	0.74	0.79	0.85	0.89
	PC	0.48	0.51	0.63	0.70	0.78	0.89	0.93
	RC	0.04	0.22	0.35	0.39	0.53	0.64	0.73
	AUC	0.08	0.31	0.51	0.55	0.62	0.74	0.82
	KA	0.01	0.09	0.21	0.32	0.48	0.64	0.74

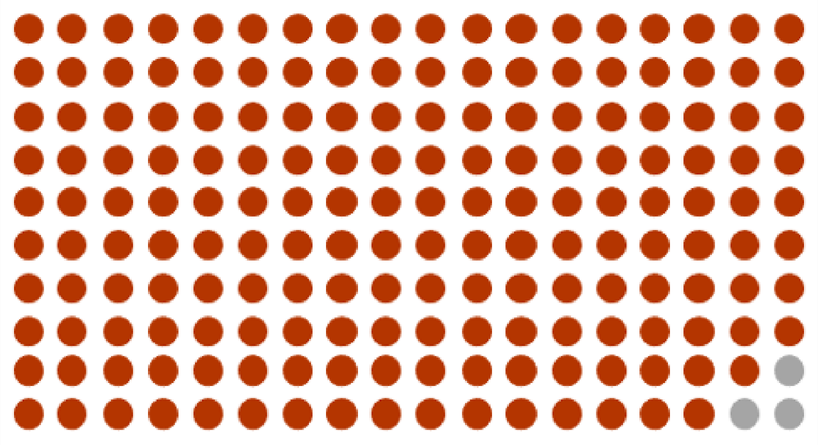
Van Zoonen, W., & Van der Meer, T. G. (2016). Social media research: The application of supervised machine learning in organizational communication research.. *Computers in Human Behavior*, *63*, 132–141.

https://doi.org/10.1016/j.chb.2016.05.028

Accuracy

Accuracy: In which percentage of all cases was our classifier right?
Class distribution: The number of examples that belong to each class.
Imbalanced classification: A classification predictive modeling problem where the distribution of examples across the classes within a training dataset is not equal.

Accuracy



Majority class (red dots) vs. minority class (grey dots)

Validating Models

Many more metrics to validate models.

Learn more using, for example, the scikit-learn documentation.

Zooming out

Today, we talked about:

- Rule-based Text Classification
- Automated Text Classification: SML
- The principles behind SML
- The steps of SML
- Some commonly used ML models
- Validating models

In this week's tutorial, you will:

- Get some hands-on experience with supervised machine learning
- Discuss the first 5 questions of the tutorial exercise of week 8

To do:

- Work on the first 5 questions of the tutorial exercise of week 8