Computational Communication Science 2
Week 6 - Lecture
»Setting Up Supervised Machine Learning«

Marthe Möller

a.m.moller@uva.nl

May 8, 2023

Digital Society Minor, University of Amsterdam

Today

Recap

Rule-based Text Classification

The principles behind SML

SML: A practical application

SML models

Recap

What we did so far

Week 1 - 3: Text as data

Week 4: Recommender systems

Use Python to set up an experiment (research project)!

What we will do next

Week 6 - 7: Automated analysis of text with Supervised Machine Learning

Use Python to analyze texts - and reflect on this!

What we will do next

Classical content analysis: Manual analysis of texts

What we will do next

This can be done automatically with Python!

Especially helpful when working with big data sets.

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.

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.

Text classification

RQ: How prevalent is flaming on Twitter?

Rule-based approach:

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

Text classification

RQ: How prevalent is flaming on Twitter?

Rule-based approach:

- Create a list with all the swearwords that exist.
- For each tweet in the dataset, use the list to count the number of swearwords
- 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."

When you simply want to count the occurence of specific words, a rule-based approach will be quick, cheap, easy, and transparent - perfect!

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."

When you simply want to count the occurence of specific words, a rule-based approach will be quick, cheap, easy, and transparent - perfect!

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."

When you simply want to count the occurence of specific words, a rule-based approach will be quick, cheap, easy, and transparent - perfect!

Text classification

Advantages of rule-based text classification:

- Simple and therefore transparant
- Cheap

Text classification

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

Text classification

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

Text classification

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.

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!

Zooming out

We talked about:

- Rule-based Text Classification
- Automated Text Classification: SML

Next, we will talk about:

- The principles behind SML
- Some commonly used SML models

The principles behind SML

The principles behind SML

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

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

x_2 = tail? (0 = no, 1 = yes)

The principles behind SML

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

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

x_2 = tail? (0 = no, 1 = yes)

The principles behind SML

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

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

x_2 = tail? (0 = no, 1 = yes)

The principles behind SML

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

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

x_2 = tail? (0 = no, 1 = 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

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

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

The principles behind SML

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

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

The principles behind SML

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

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

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

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

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

Traditional usage of models in CS: to explain

Usage of models in ML: to predict

The principles behind SML

Traditional usage of models in CS: to explain

Usage of models in ML: to predict

The principles behind SML

Compare:

RQ: To what extent does the amount of hours spend playing violent video games predict aggressive behavior by individuals?

RQ: Given the amount of hours that an individual spents playing violen video games, how likely is this person to show aggressive behavior?

The principles behind SML

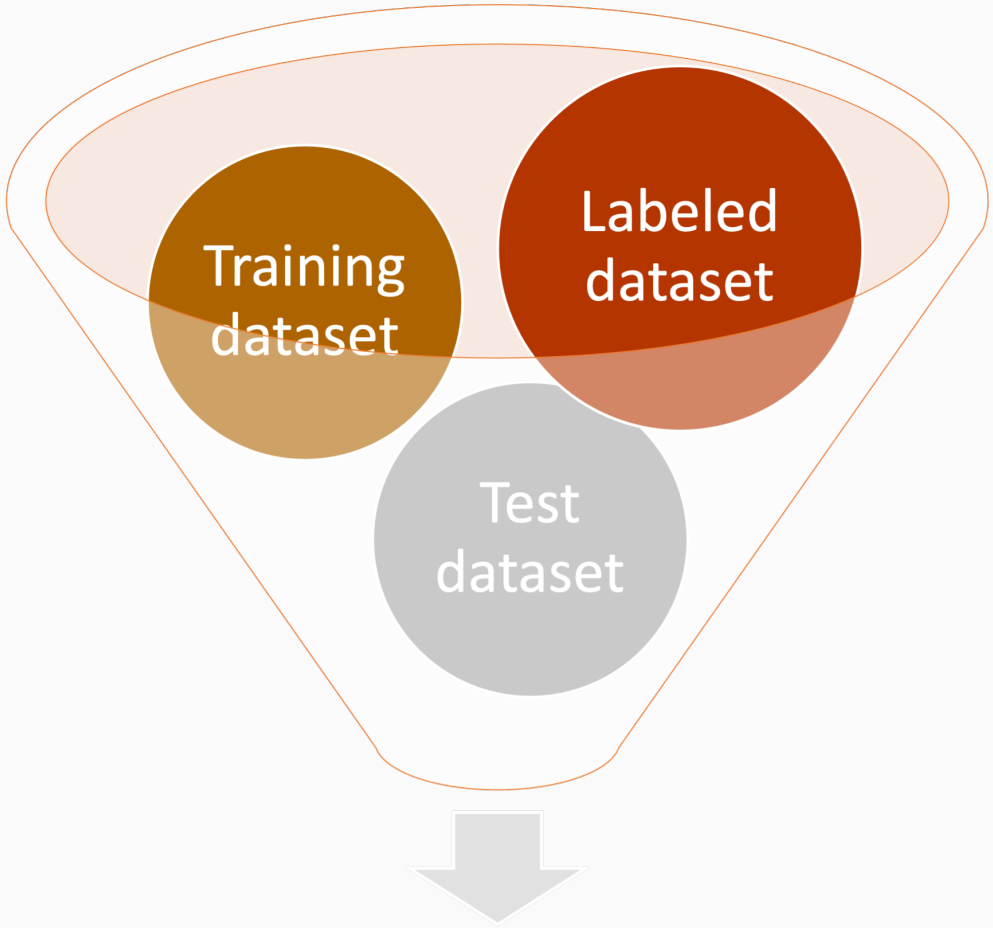
Can you think of an example case where SML can be useful?

The principles behind SML

You know know about the principles of SML.

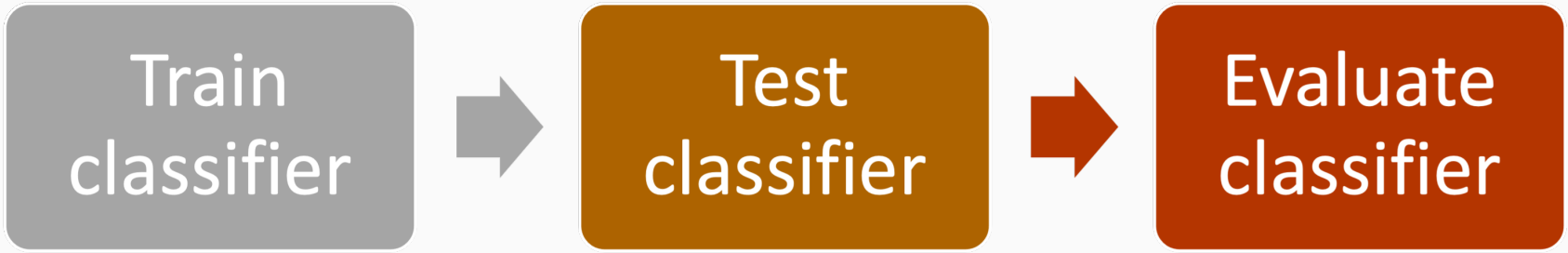
What does the process of SML look like?

SML step by step



Machine Learning Process

SML step by step



SML step by step

Today, more about the first step!

(Next week, more about the second and last step)

Zooming out

We talked about:

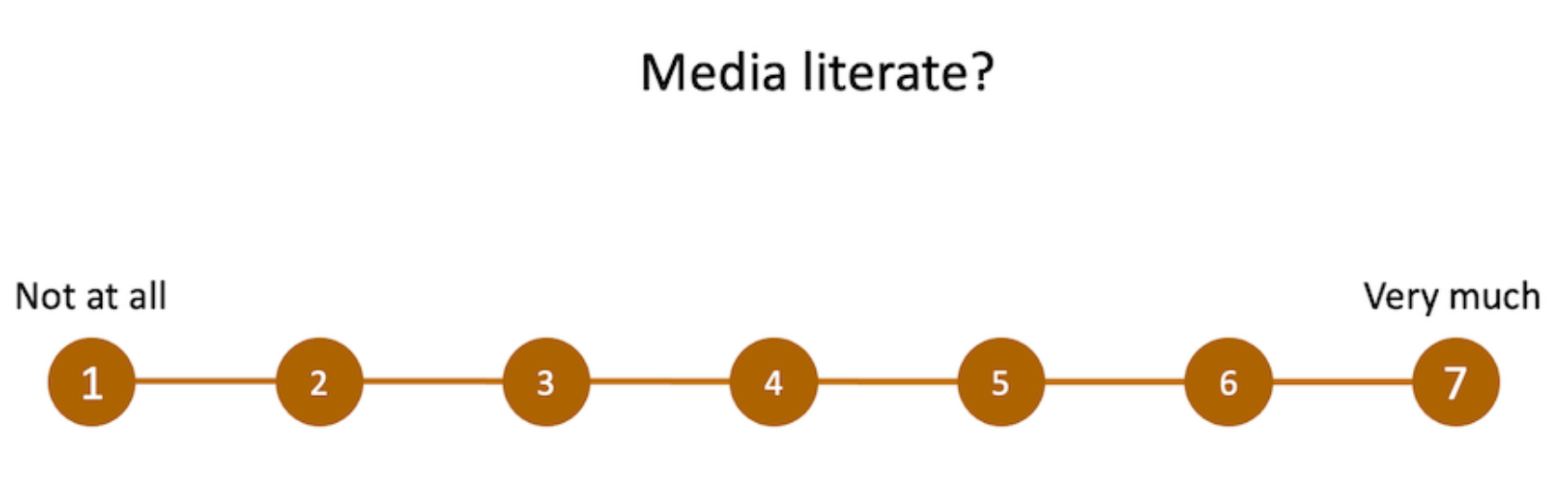
- Rule-based Text Classification
- Automated Text Classification: SML
- The principles behind SML

Next, we will talk about:

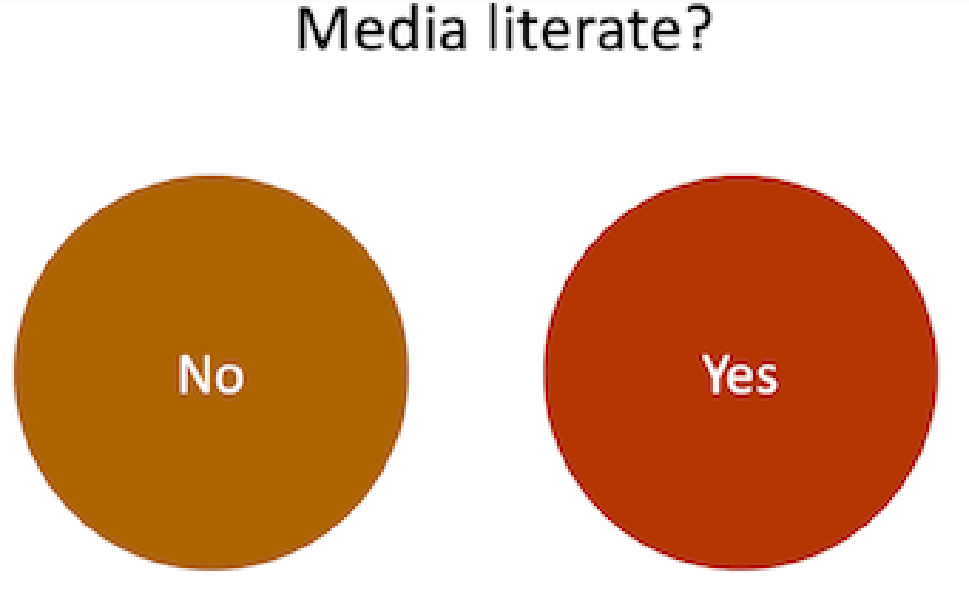
- A practical application of SML
- Some commonly used SML models

SML: A practical application

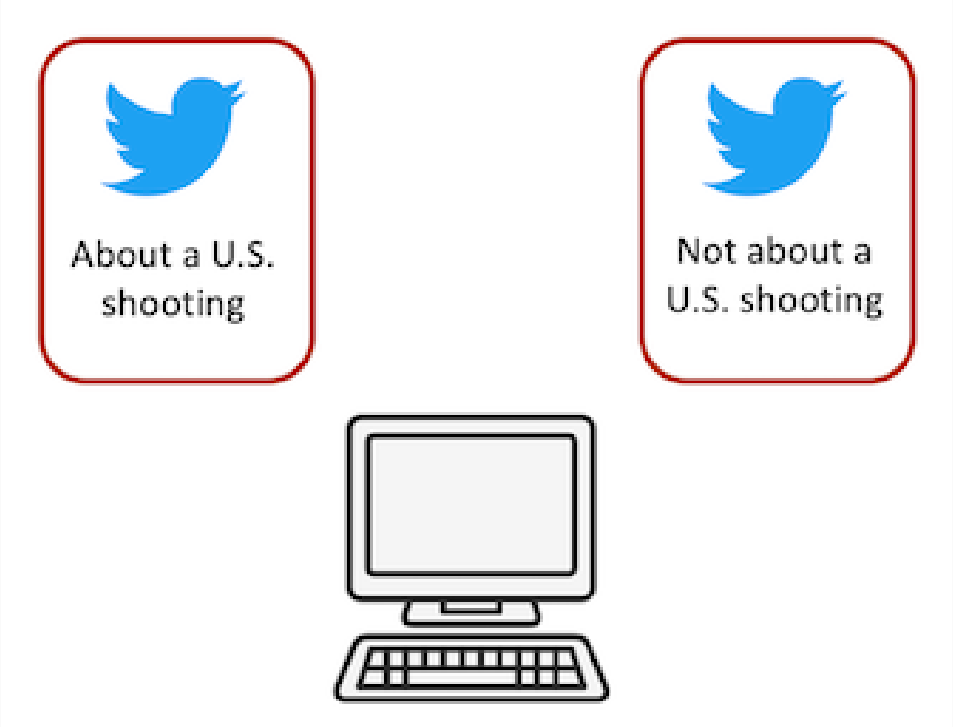
Regression



Logistic Regression



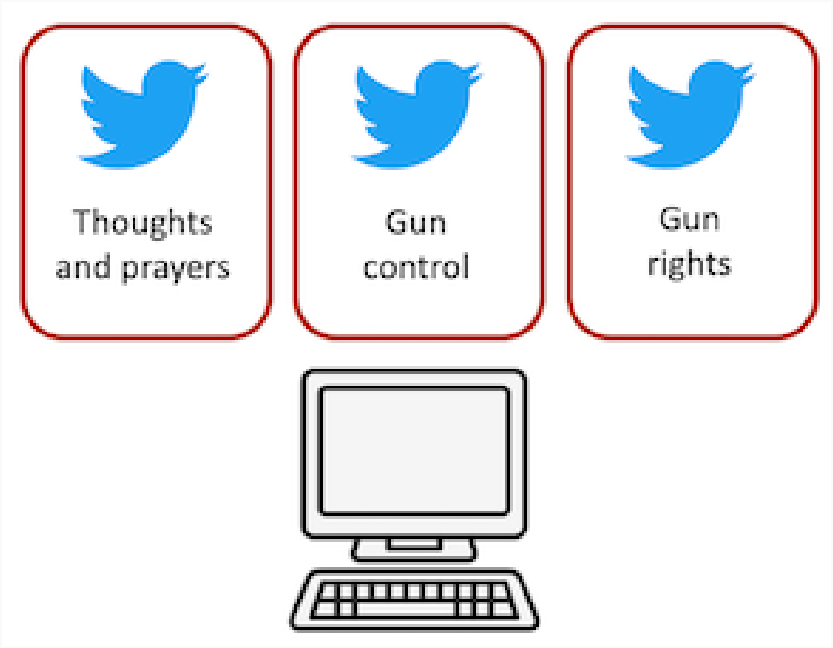
Logistic Regression



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

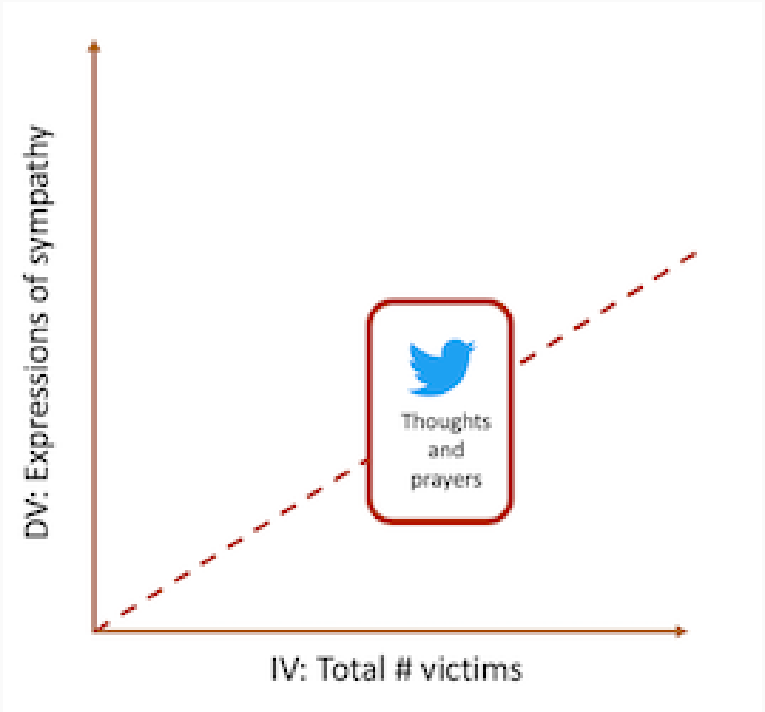
Logistic Regression



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:

```
1  import csv
2  from sklearn.model_selection import train_test_split
3
4  tweets = []
5  labels = []
6
7  with open(file) as fi:
8    data = csv.reader(fi, delimiter='\t')
9    for row in data:
10     tweets.append(row[0])
11     labels.append(row[1])
12
13 tweets_train, tweets_test, y_train, y_test = train_test_split(tweets,
       labels, test_size=0.2, random_state=42)
```

What does this look like in code?

Four lists:

```
1
2 tweets_train = ['Tweet about shooting', 'Another tweet', '
      Shooting!']
3 tweets_test = ['One more tweet']
4
5 y_train = [1, 0, 1]
6 y_test = [0]
```

What does this look like in code?

Second, vectorize the texts that need to be labeled:

```
1 from sklearn.feature_extraction.text import (TfidfVectorizer)
2
3 tfidfvectorizer = TfidfVectorizer(stop_words="english")
4 X_train = tfidfvectorizer.fit_transform(tweets_train)
5 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:

```
1 from sklearn.linear_model import (LogisticRegression)
2
3 logres = LogisticRegression()
4 logres.fit(X_train, labels_train)
5 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:

```
1  from sklearn.feature_extraction.text import (TfidfVectorizer)
2  from sklearn.linear_model import (LogisticRegression)
3
4  tfidfvectorizer = TfidfVectorizer(stop_words="english")
5  X_train = tfidfvectorizer.fit_transform(tweets_train)
6  X_test = tfidfvectorizer.transform(tweets_test)
7
8  logres = LogisticRegression()
9  logres.fit(X_train, labels_train)
10 y_pred = logres.predict(X_test)
```

SML models

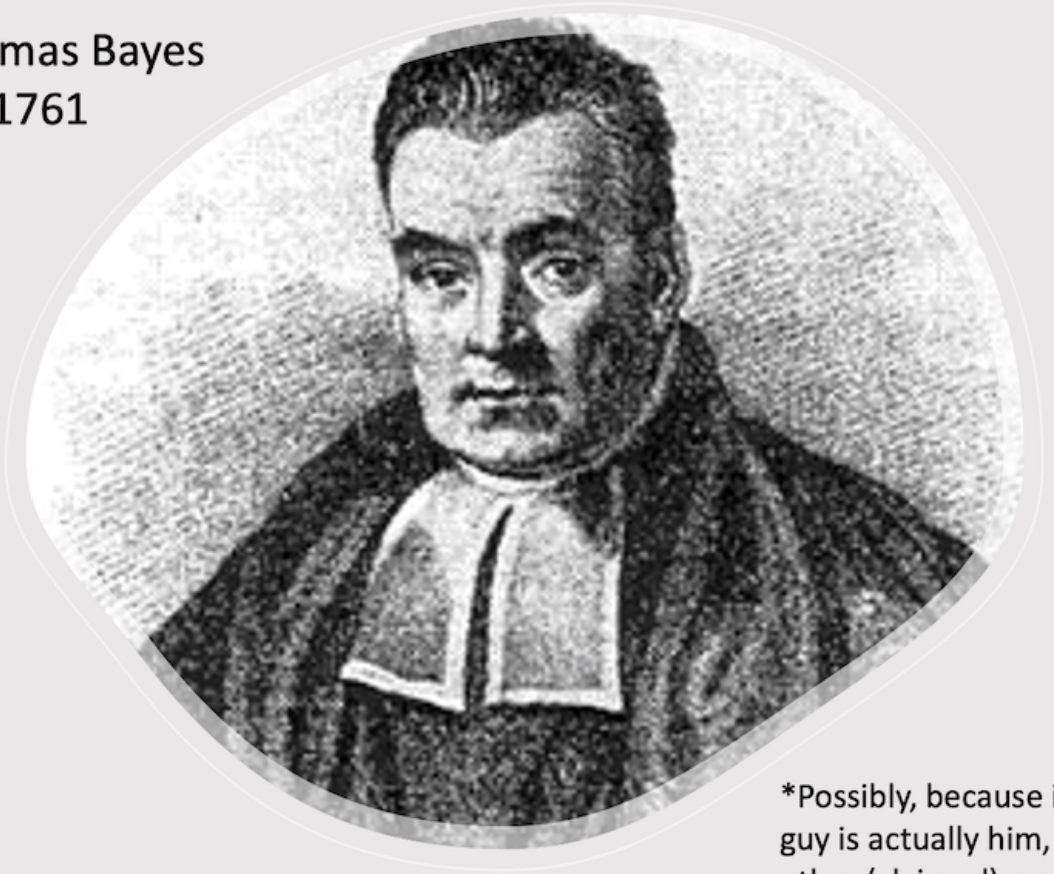
Logistic Regression is one commonly used model to train classifiers.

Let's talk about other models as well!

SML models

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

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

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

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

What does this look like in code?

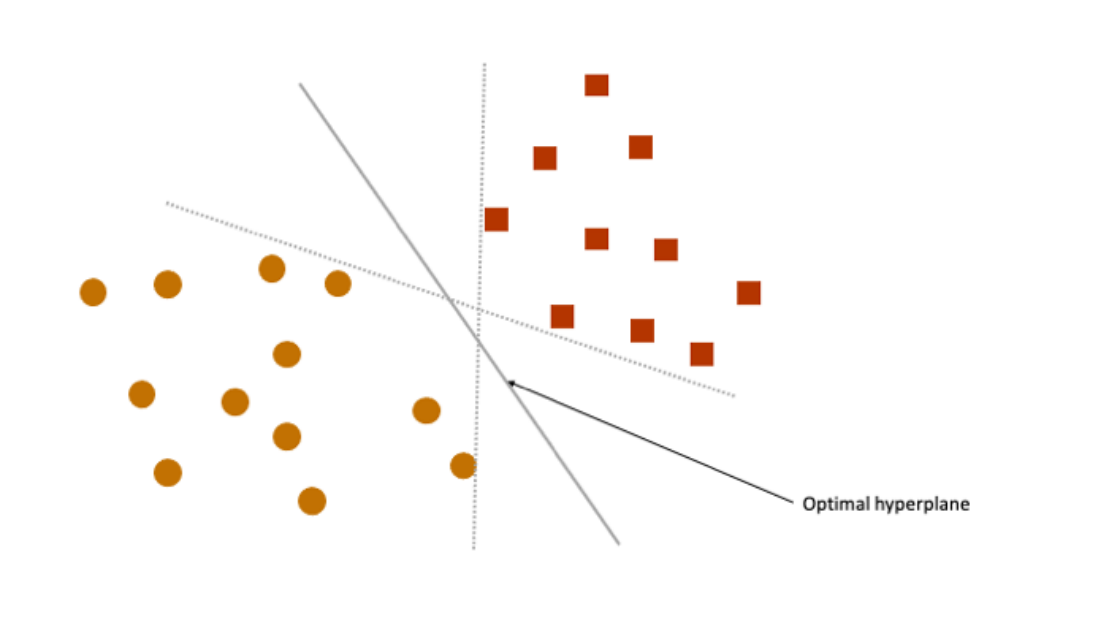
Let's also train a model based on a count vectorizer and Naïve Bayes:

```
1  from sklearn.feature_extraction.text import (CountVectorizer)
2  from sklearn.naive_bayes import MultinomialNBB
3
4  countvectorizer = CountVectorizer(stop_words="english")
5  X_train = countvectorizer.fit_transform(texts_train)
6  X_test = countvectorizer.transform(texts_test)
7
8  nb = MultinomialNB()
9  nb.fit(X_train, labels_train)
10 y_pred = nb.predict(X_test)
```

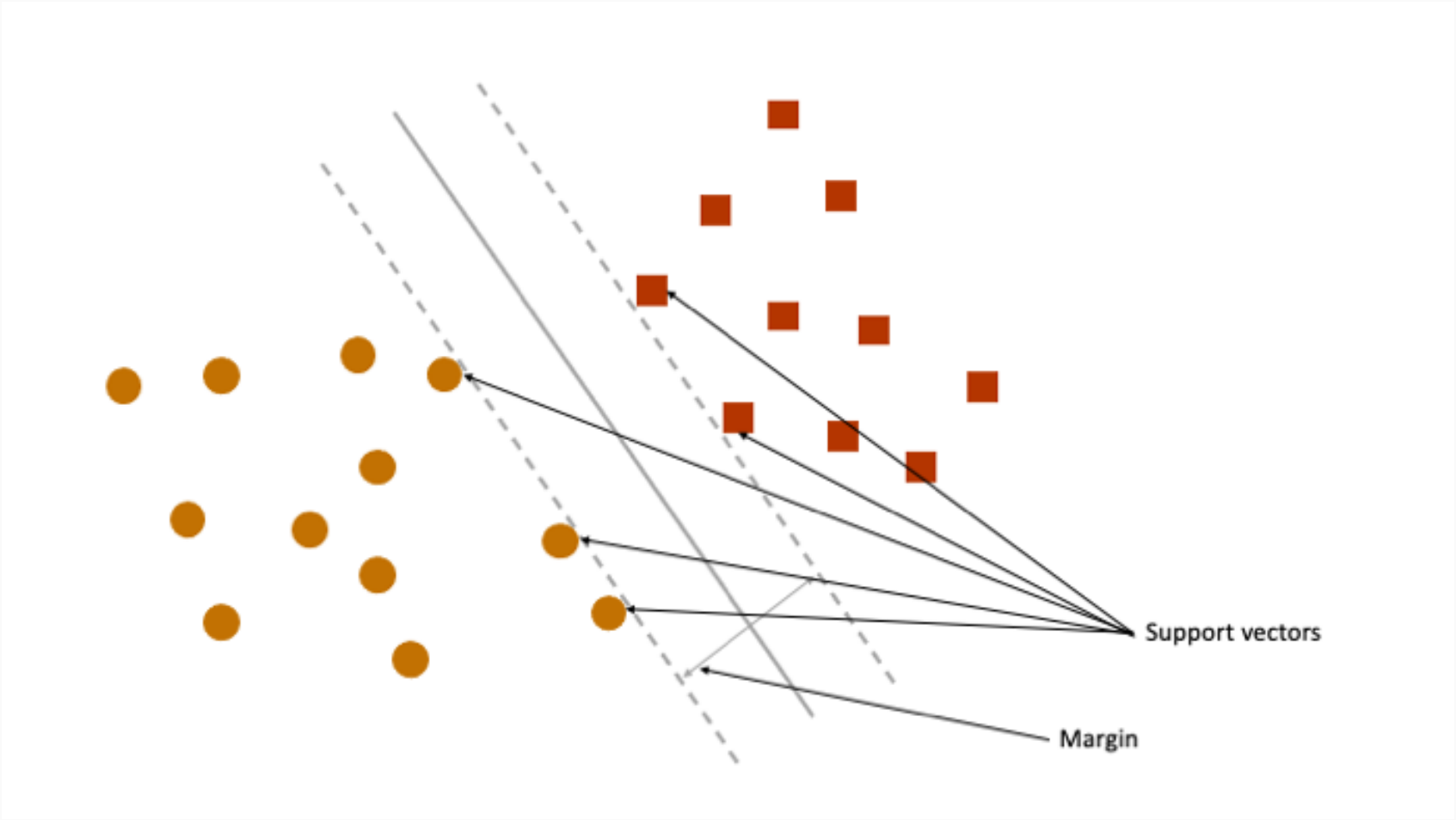
Support Vector Machines

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

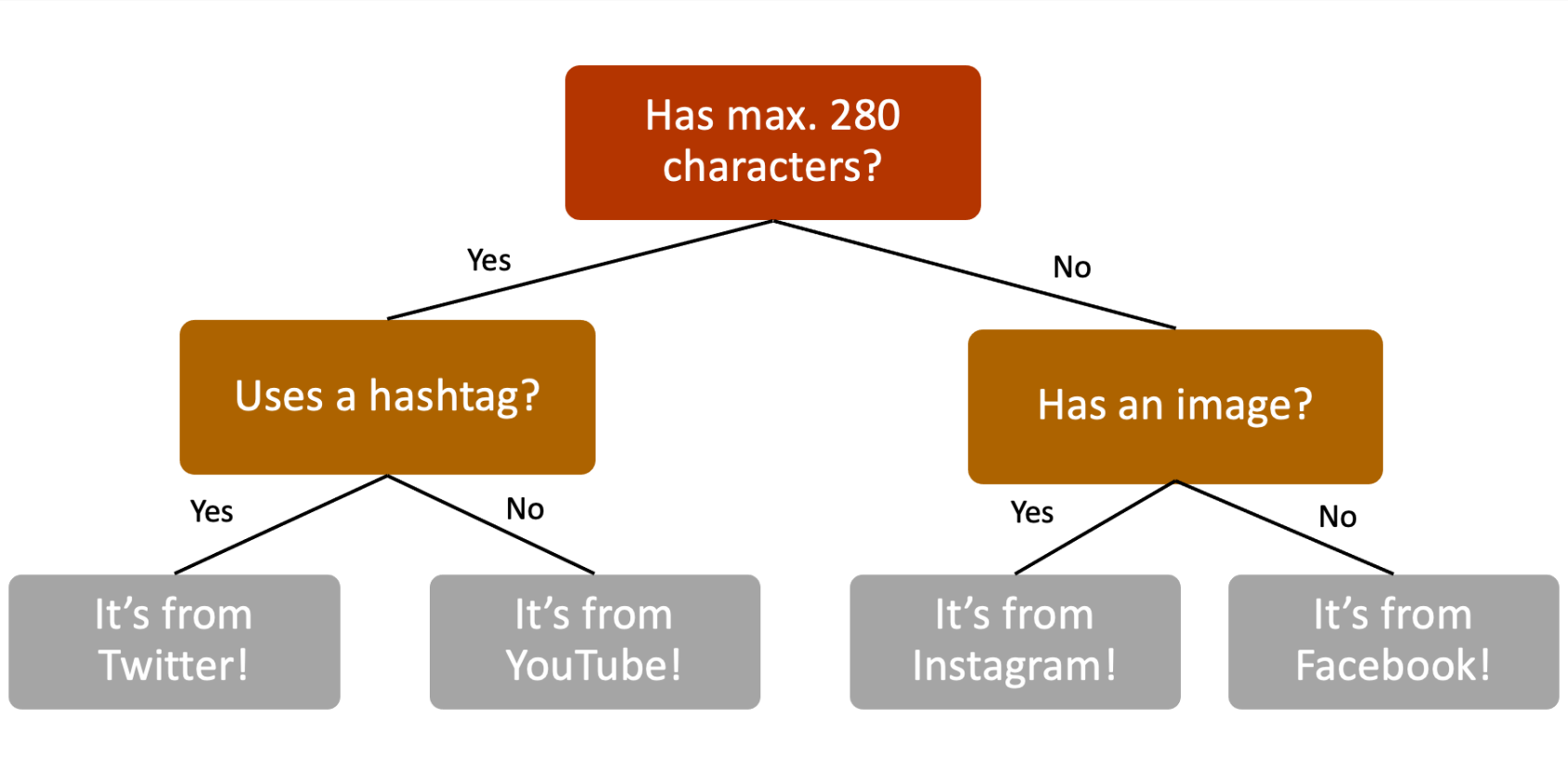
The best hyperplane is the one that has the maximum margin (distance) between the datapoints of both classes.



Support Vector Machines



Decision Trees and Random Forests



Decision Trees and Random Forests

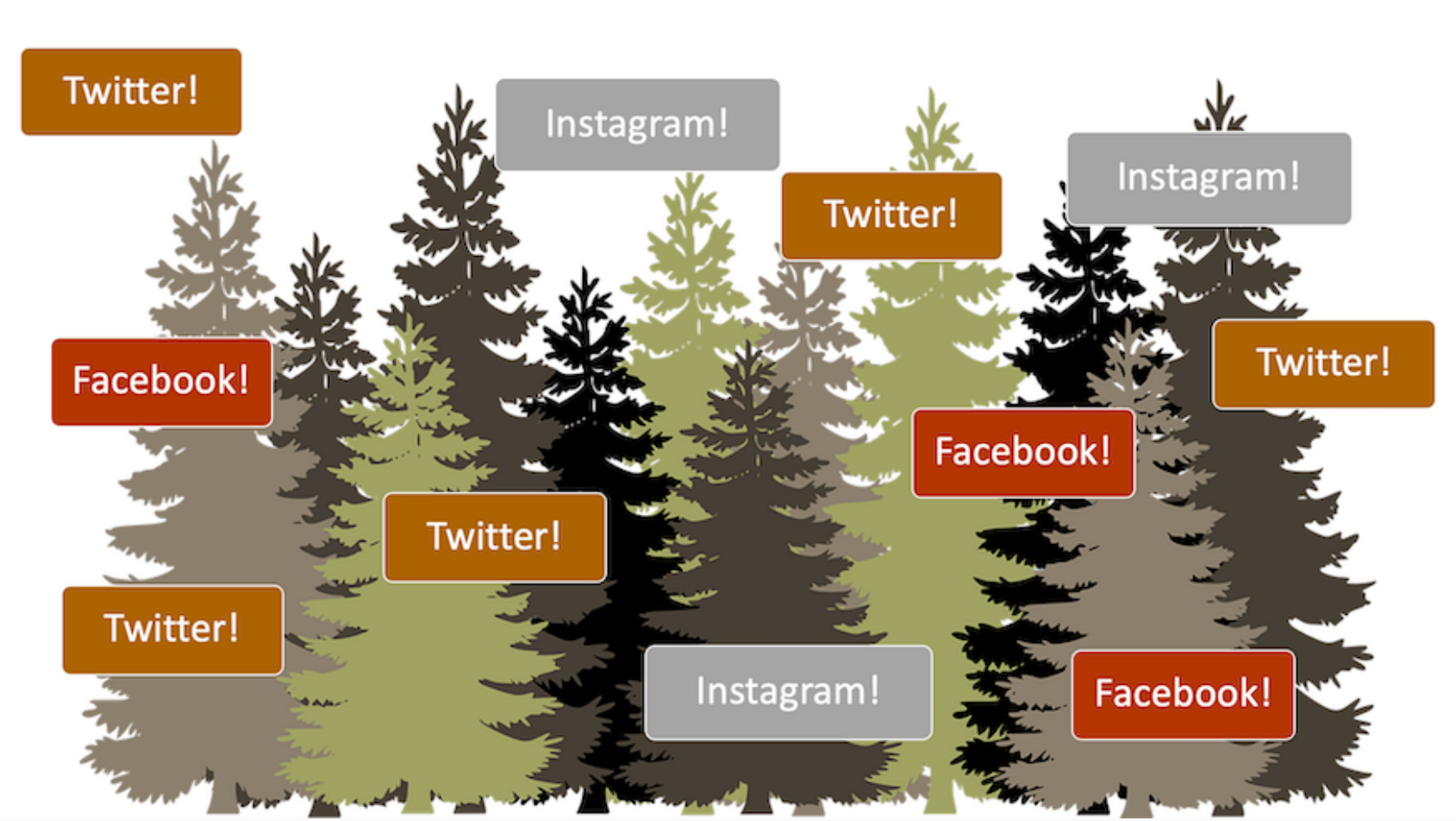
Advantages of decision trees:

- 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

Today, we talked about:

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

Zooming out

Tomorrow and this week, you will:

- Get some hands-on experience with supervised machine learning!

Work on the the tutorial exercises for this week.

Refs

References

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

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

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