gesis Leibniz-Institut für Sozialwissenschaften

Multi-class and Multi-label Text Classification in Python

Lukas Birkenmaier

Workshop for Ukraine, 22.02.2024

Leibniz Gemeinschaft

About me

RWTH AACHEN UNIVERSITY

Universität Konstanz

UNIVERSITAS WARWICENSIS

Maastricht University

gesis Leibniz Institute for the Social Sciences

pwc

Baden-Württemberg
MINISTERIUM FÜR WIRTSCHAFT, ARBEIT UND WOHNUNGSBAU

UNIVERSITÄT DUISBURG ESSEN
Offen im Denken

gesis Leibniz-Institut für Sozialwissenschaften

Agenda

- Introduction
- Classification Basics
  - Machine Learning
  - Validation (Accuracy, Precision, Recall, F1 Score)
- **Multi-class** Classification
  - Model Architecture
  - Validation
  - Live-Coding
- **Multi-label** Classification
  - Model Architecture
  - Validation
  - Live-Coding

Disclaimer

- This is an applied course!
  - Some knowledge of Python is useful. However, you can run the scripts and interpret the output without any python knowledge
  - No mathematical deep-dive, focus on applied setting
  - You are invited to go your own pace (e.g., adapting the scripts to a new dataset)

! The materials are designed so that you can look at them later and copy / paste certain code snippets for your own work!

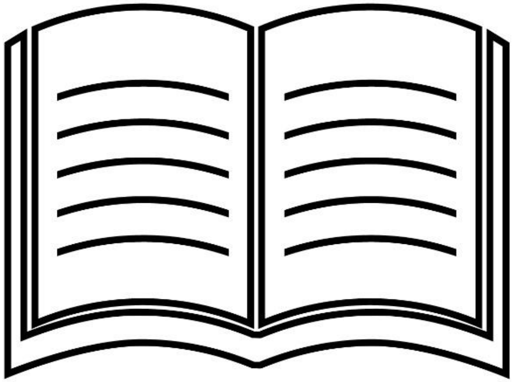
By the end of this course, you will have…

- **learned the basic terminology** around text classification
- **understood the difference** between **multi-class and multi-label classification**
- **learned** the fundamental strategies to **evaluate** classification tasks
- have **applied two classification tasks** using python code
- had some evening fun ☺

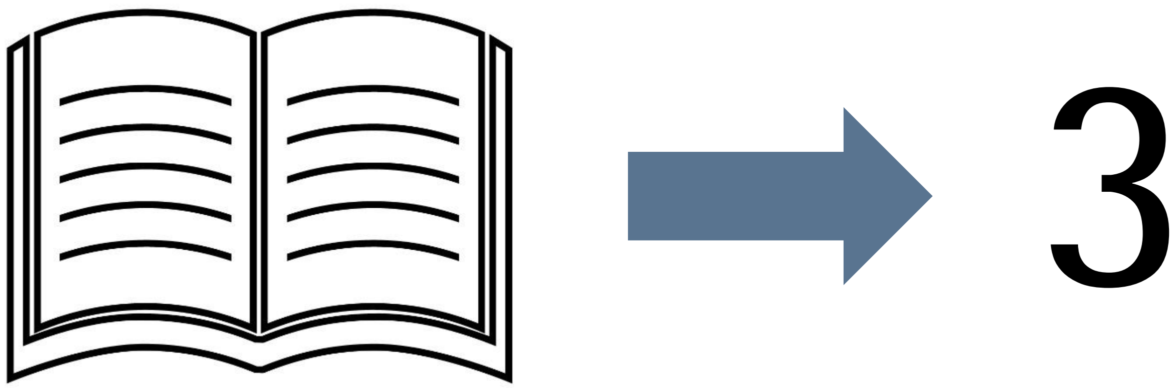
Introduction

gesis Leibniz-Institut für Sozialwissenschaften

Text Classification

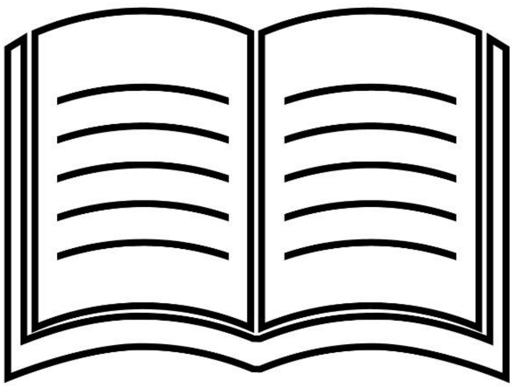


Text Classification



Text Classification

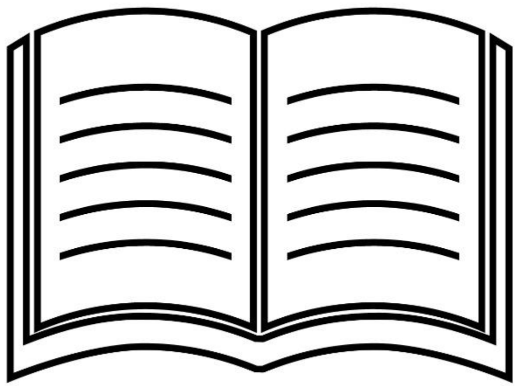
"Classification is the process of accurately classifying previously undiscovered data"



3

Text Classification

“Classification is the process of accurately classifying previously undiscovered data”



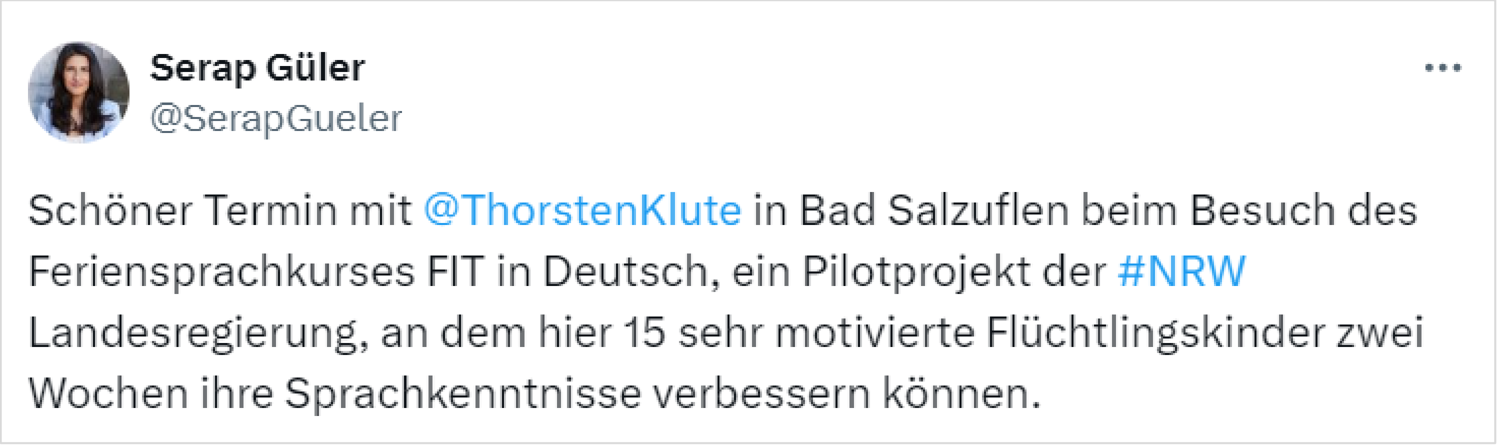
3

Many practical applications, e.g.,

- **Spam filtering**
- **Hate Speech Detection**
- **Language Identification**
- **Policy Issue Classification**
- **Sentiment Analysis**

Practical Application: Policy Issue

- "Great appointment with @ThorstenKlute in Bad Salzuflen during the visit of the holiday language course FIT in German, a pilot project of the #NRW state government, where 15 very motivated refugee children can improve their language skills for two weeks."



Serap Güler
@SerapGueler

Schöner Termin mit @ThorstenKlute in Bad Salzuflen beim Besuch des Feriensprachkurses FIT in Deutsch, ein Pilotprojekt der #NRW Landesregierung, an dem hier 15 sehr motivierte Flüchtlingskinder zwei Wochen ihre Sprachkenntnisse verbessern können.

Practical Application: Policy Issue

- "The city of Gütersloh prefers migrants in the allocation of building plots & explicitly wishes for a 'social and multicultural mixing' of certain areas... 🤔 😐 This 'mixing' has worked out well so far in Germany..."

Torben Braga
@torben_braga
...

Die Stadt Gütersloh bevorzugt Migranten bei der Vergabe von Baugrundstücken & wünscht sich ausdrücklich "eine soziale und multikulturelle Durchmischung" bestimmter Gebiete... 🤦‍♂️ 🙄
Hat ja bisher gut funktioniert, diese "Durchmischung" in Deutschland...

guetersloh.de/de-wAssets/doc...

gesis Leibniz-Institut für Sozialwissenschaften

Practical Application: Policy Issue

domesticsecurity	economy	education	environment	event	foreign_policy	healthcare	immigration_asylum	infrastru
0.00846947	0.01213120	0.00303444	0.00365524	0.08787690	0.00814045	0.00984293	0.00361202	0.0
0.49173900	0.00862222	0.02396040	0.00722634	0.02777380	0.06010910	0.04110670	0.02433700	0.0
0.00207624	0.84491000	0.00554972	0.02222480	0.00462421	0.00792736	0.01402210	0.00224846	0.0
0.22862300	0.01242660	0.03978020	0.01281590	0.02577220	0.11836900	0.06211580	0.14245400	0.0
0.00511126	0.03454080	0.00846039	0.83044300	0.00420553	0.00625391	0.00853137	0.00338545	0.0
0.00511126	0.03454080	0.00846039	0.83044300	0.00420553	0.00625391	0.00853137	0.00338545	0.0
0.00435295	0.01014710	0.00135177	0.00442443	0.12319500	0.00590515	0.00495645	0.00260760	0.0
0.00702387	0.01145560	0.00168932	0.00467323	0.14338500	0.00775695	0.00697157	0.00317667	0.0
0.00796364	0.01158170	0.00564254	0.01338690	0.00484134	0.00790940	0.00498853	0.00672930	0.0
0.00470961	0.00748688	0.00133547	0.00365910	0.23994900	0.00369224	0.00488763	0.00210484	0.0
0.01782000	0.03557480	0.02762700	0.00729218	0.03514410	0.00532206	0.68964800	0.01338570	0.0
0.00465674	0.01040610	0.00125193	0.00424651	0.16369100	0.00526372	0.00579288	0.00251573	0.0
0.00620032	0.01251060	0.00180104	0.00453758	0.11416200	0.00662017	0.00665563	0.00304615	0.0
0.39065300	0.00464934	0.01538650	0.00697752	0.01480330	0.32360300	0.02635410	0.06719360	0.0
0.00552299	0.00635472	0.00126635	0.00347297	0.30504100	0.00373231	0.00436245	0.00197498	0.0
0.03917740	0.02858820	0.21272500	0.00915515	0.06832960	0.10566500	0.07544480	0.03515130	0.0
0.00752966	0.00798649	0.00198276	0.00385448	0.27171200	0.00458332	0.00559928	0.00233327	0.0
0.00613405	0.01431040	0.00191910	0.00552868	0.08412830	0.00834662	0.00627282	0.00285583	0.0

gesis Leibniz-Institut für Sozialwissenschaften

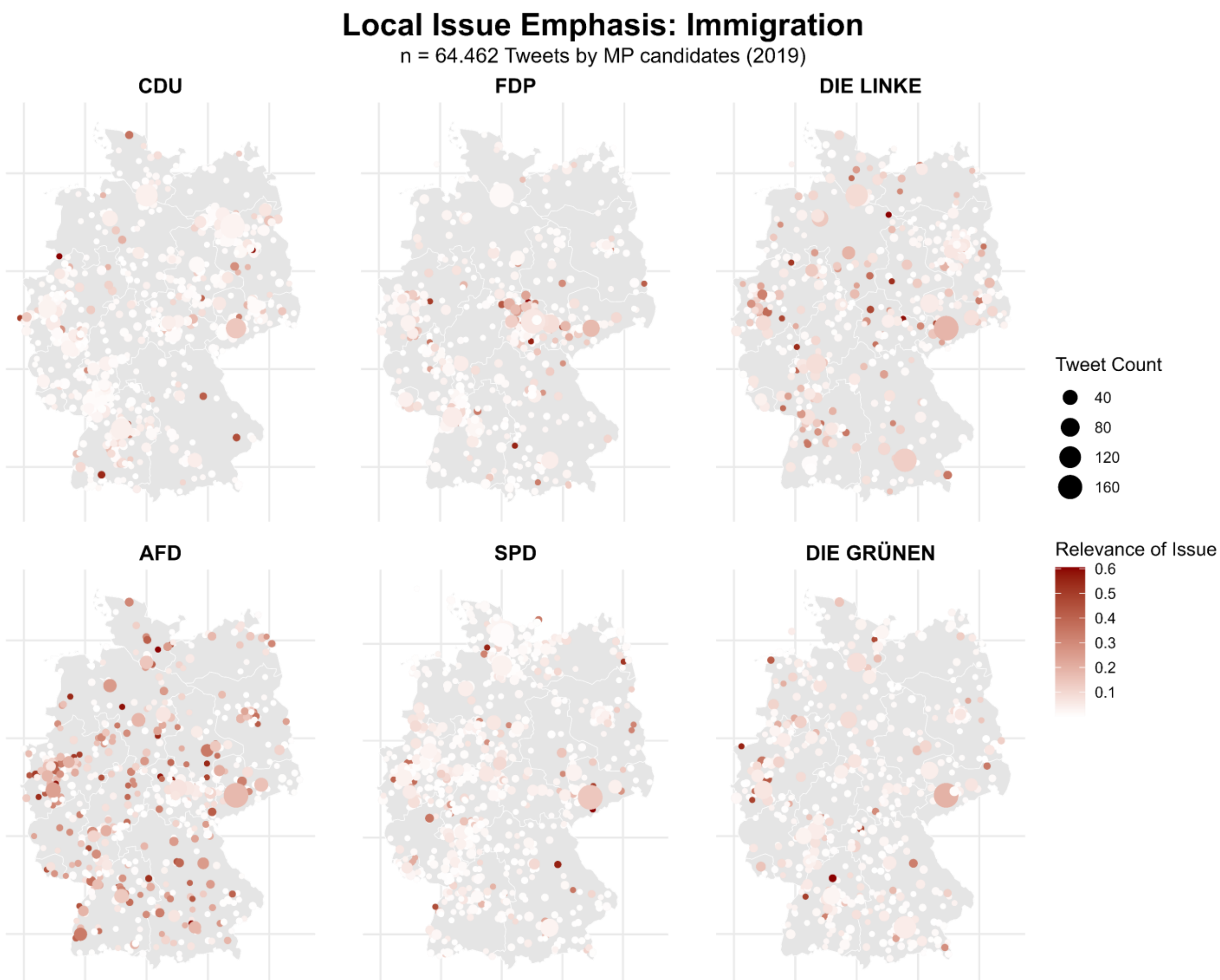
Practical Application: Policy Issue

In the dataset, each row represents a Tweet that mentions a place (e.g., a city or a region)

Probabilities for belonging to the topic "Immigration"

mesticsecurity	economy	education	environment	event	foreign_policy	healthcare	immigration_asylum	infrastru
0.00846947	0.01213120	0.00303444	0.00365524	0.08787690	0.00814045	0.0098429	0.00361202	0.0
0.49173900	0.00862222	0.02396040	0.00722634	0.02777380	0.06010910	0.0411067	0.02433700	0.0
0.00207624	0.84491000	0.00554972	0.02222480	0.00462421	0.00792736	0.0140221	0.00224846	0.0
0.22862300	0.01242660	0.03978020	0.01281590	0.02577220	0.11836900	0.0621158	0.14245400	0.0
0.00511126	0.03454080	0.00846039	0.83044300	0.00420553	0.00625391	0.0085313	0.00338545	0.0
0.00511126	0.03454080	0.00846039	0.83044300	0.00420553	0.00625391	0.0085313	0.00338545	0.0
0.00435295	0.01014710	0.00135177	0.00442443	0.12319500	0.00590515	0.0049564	0.00260760	0.0
0.00702387	0.01145560	0.00168932	0.00467323	0.14338500	0.00775695	0.0069715	0.00317667	0.0
0.00796364	0.01158170	0.00564254	0.01338690	0.00484134	0.00790940	0.0049885	0.00672930	0.0
0.00470961	0.00748688	0.00133547	0.00365910	0.23994900	0.00369224	0.0048876	0.00210484	0.0
0.01782000	0.03557480	0.02762700	0.00729218	0.03514410	0.00532206	0.6896480	0.01338570	0.0
0.00465674	0.01040610	0.00125193	0.00424651	0.16369100	0.00526372	0.0057928	0.00251573	0.0
0.00620032	0.01251060	0.00180104	0.00453758	0.11416200	0.00662017	0.0066556	0.00304615	0.0
0.39065300	0.00464934	0.01538650	0.00697752	0.01480330	0.32360300	0.0263541	0.06719360	0.0
0.00552299	0.00635472	0.00126635	0.00347297	0.30504100	0.00373231	0.0043624	0.00197498	0.0
0.03917740	0.02858820	0.21272500	0.00915515	0.06832960	0.10566500	0.0754448	0.03515130	0.0
0.00752966	0.00798649	0.00198276	0.00385448	0.27171200	0.00458332	0.0055992	0.00233327	0.0
0.00613405	0.01431040	0.00191910	0.00552868	0.08412830	0.00834662	0.0062728	0.00285583	0.0

Practical Application: Policy Issue and Places*

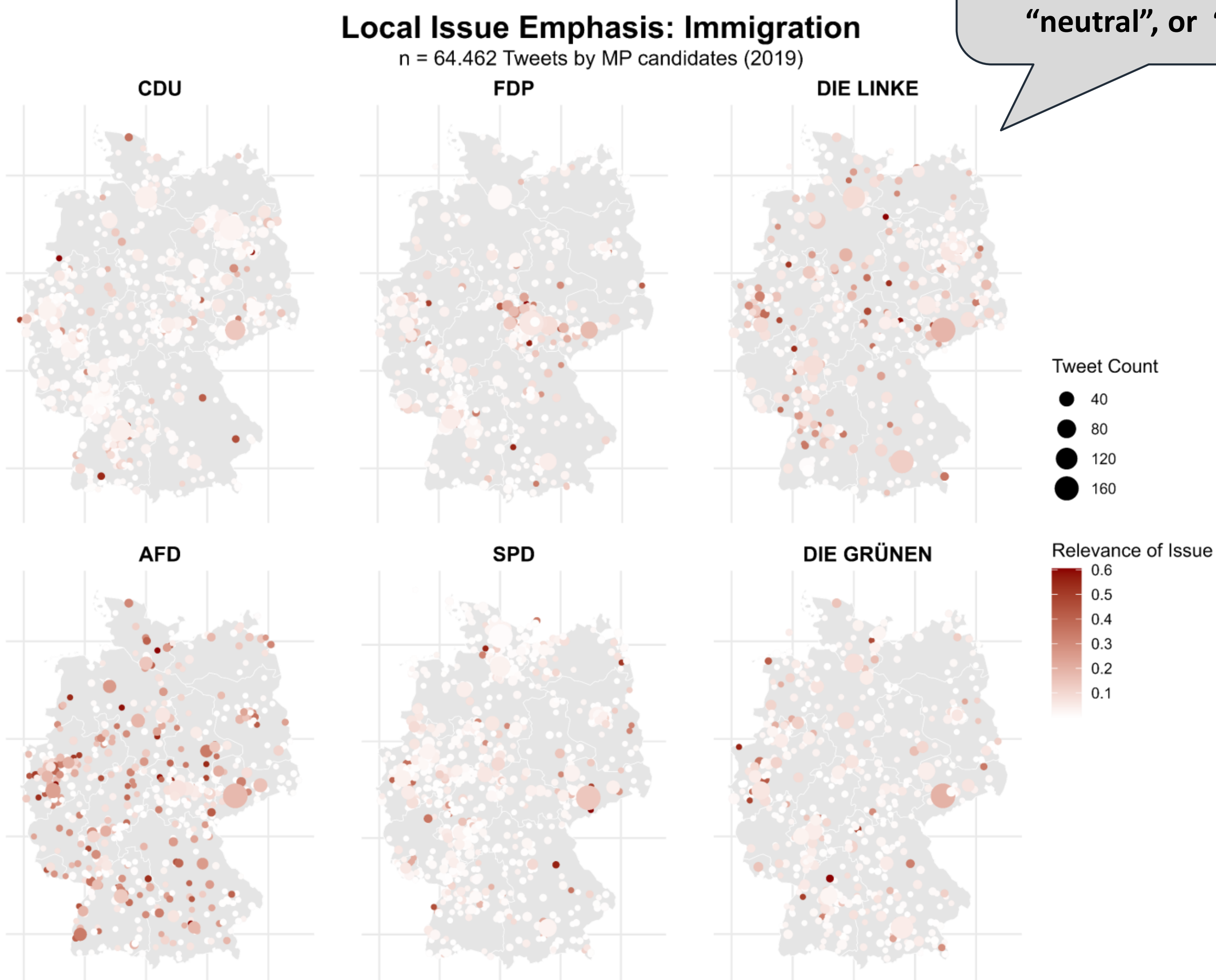


*Ongoing work

gesis Leibniz-Institut für Sozialwissenschaften

Practical Application: Policy Issue and

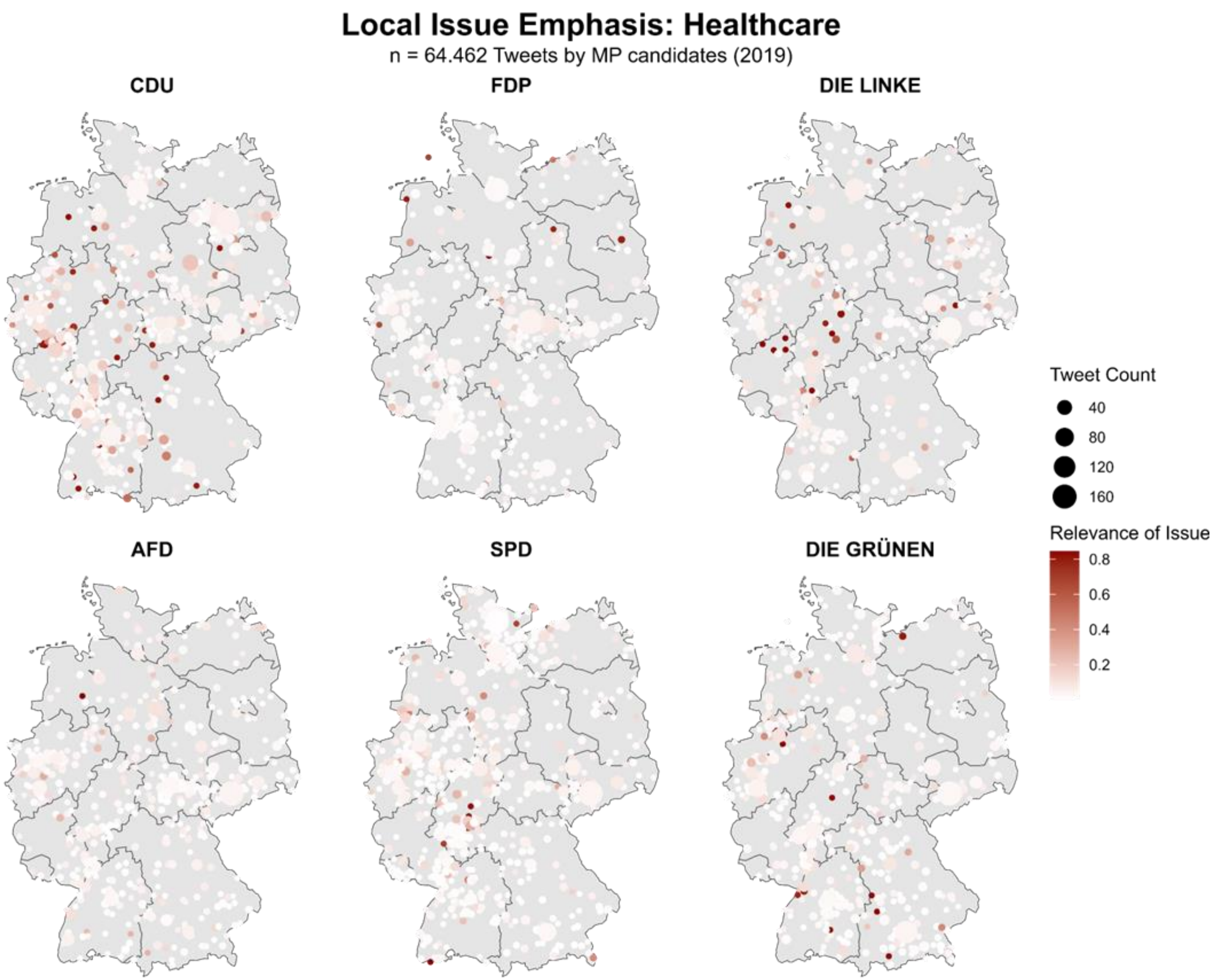
We could further apply sentiment analysis to classify the tweets into “negative”, “neutral”, or “positive”



*Ongoing work

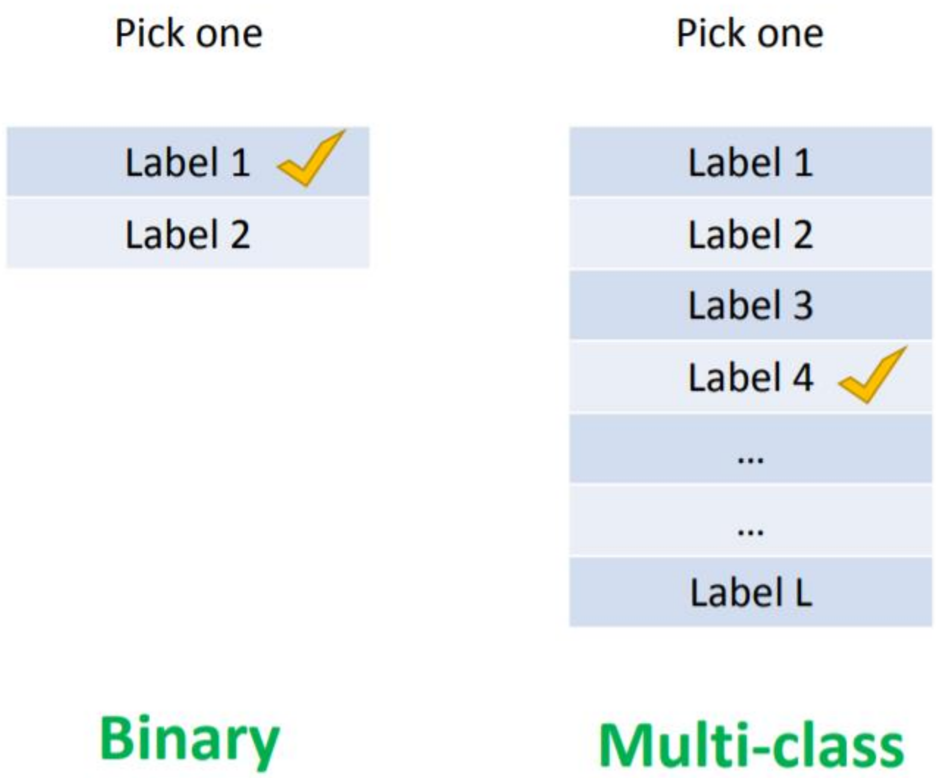
gesis Leibniz-Institut für Sozialwissenschaften

Practical Application: Policy Issue and Places*



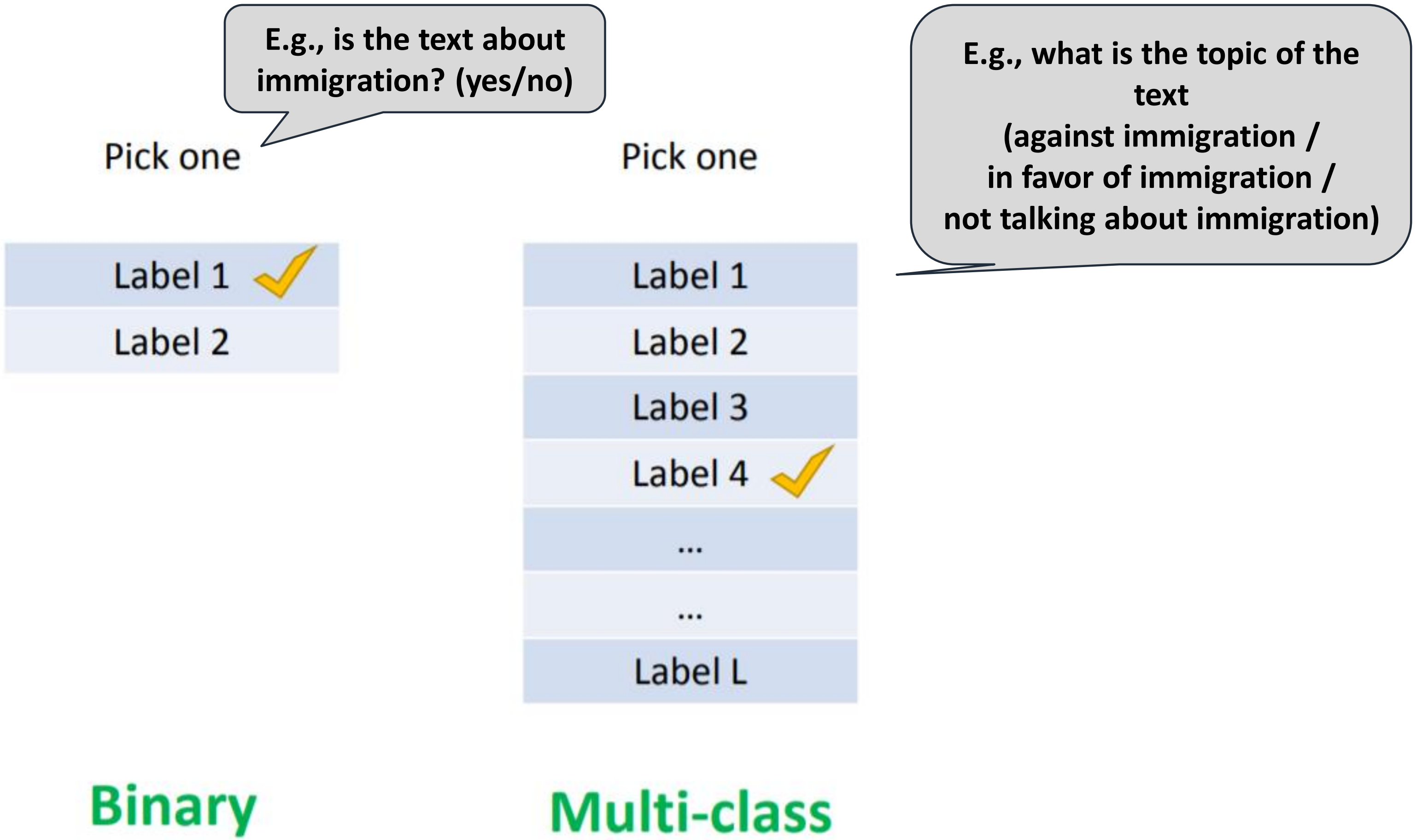
*Ongoing work

General Overview Classification



https://harshjadhav100.medium.com/multiclass-classification-vs-multi-label-classification-a37cbe1627e

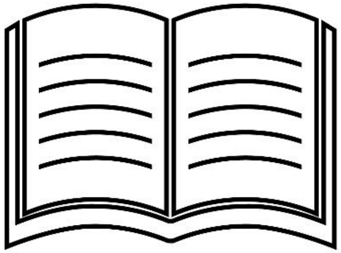
General Overview Classification



https://harshjadhav100.medium.com/multiclass-classification-vs-multi-label-classification-a37cbe1627e

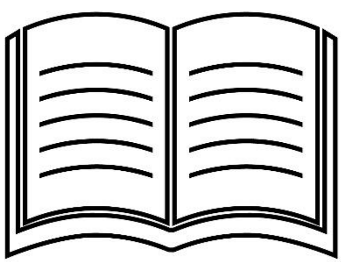
Text Classification

"Digital Technologies can help children to understand our ecosystem better"



Text Classification

"Digital Technologies can help children to understand our ecosystem better"

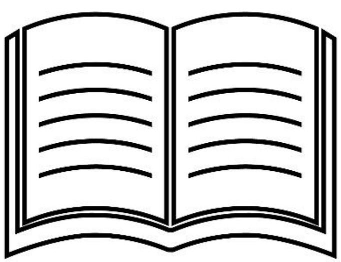


Topic: Environment?

gesis Leibniz-Institut für Sozialwissenschaften

Text Classification

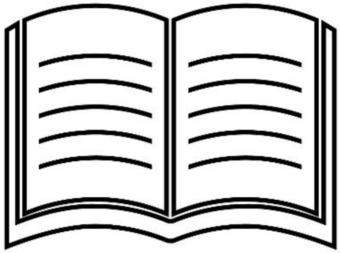
"Digital Technologies can help children to understand our ecosystem better"



Topic: Digitization?

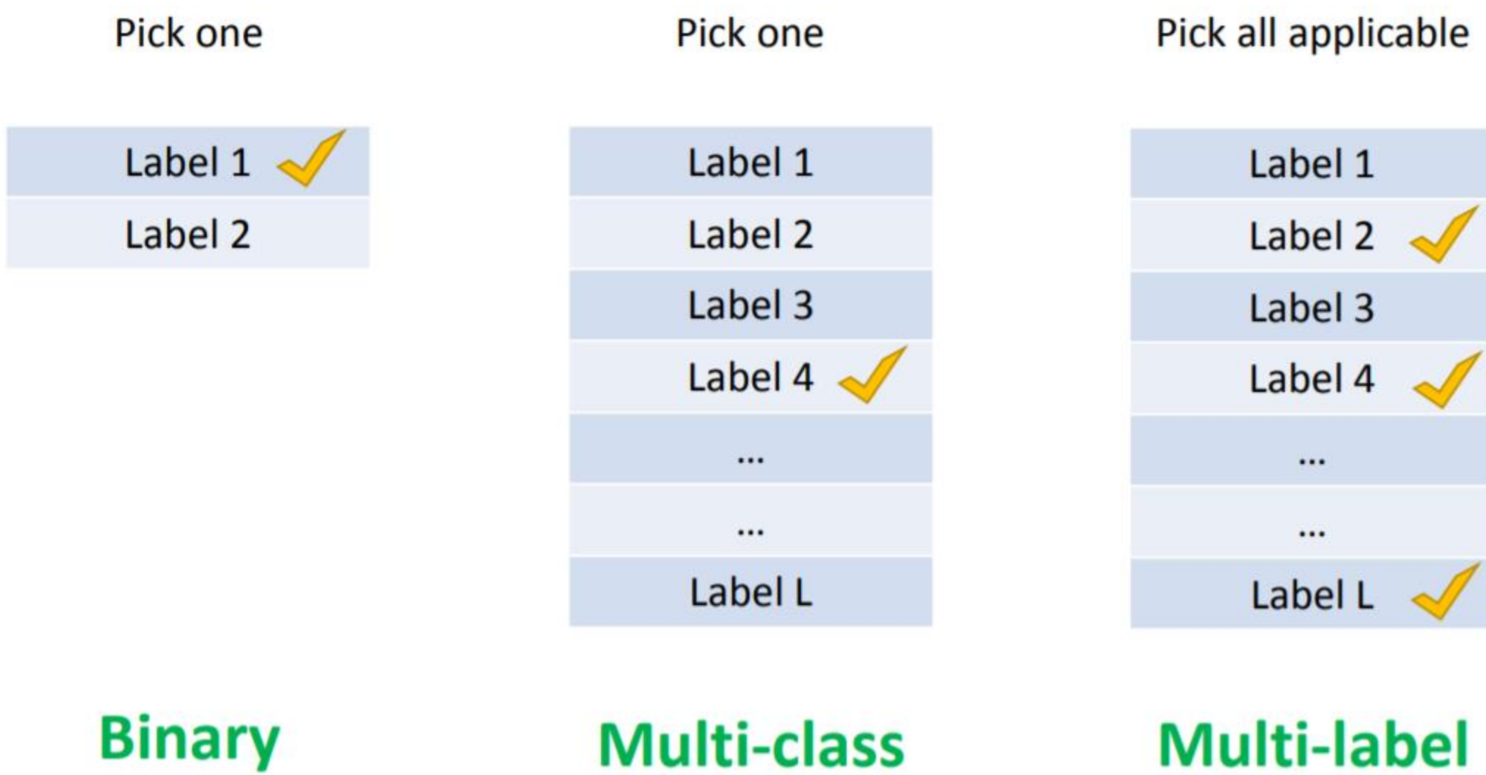
Text Classification

"Digital Technologies can help children to understand our ecosystem better"



Topic: Education?

General Overview Classification



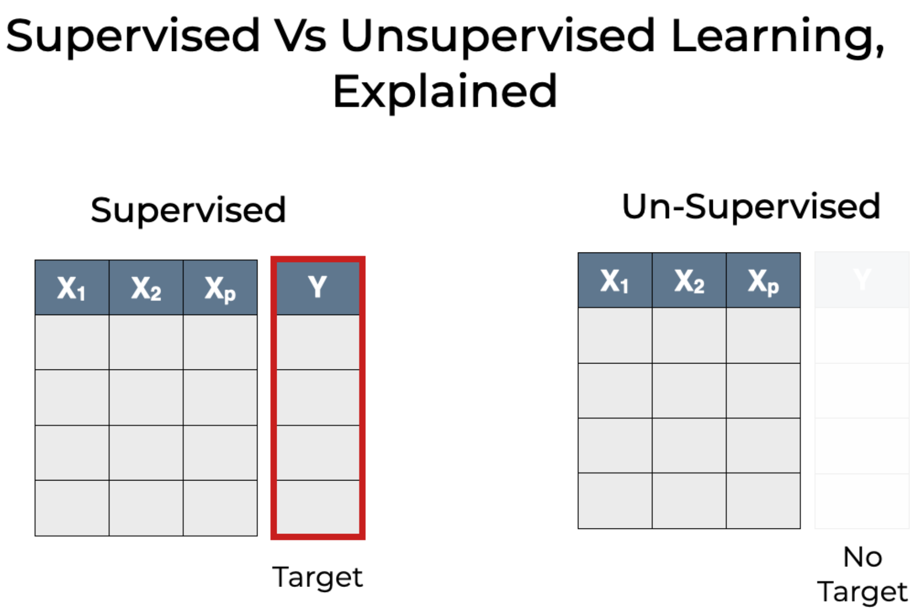
https://harshjadhav100.medium.com/multiclass-classification-vs-multi-label-classification-a37cbe1627e

Classification Basics

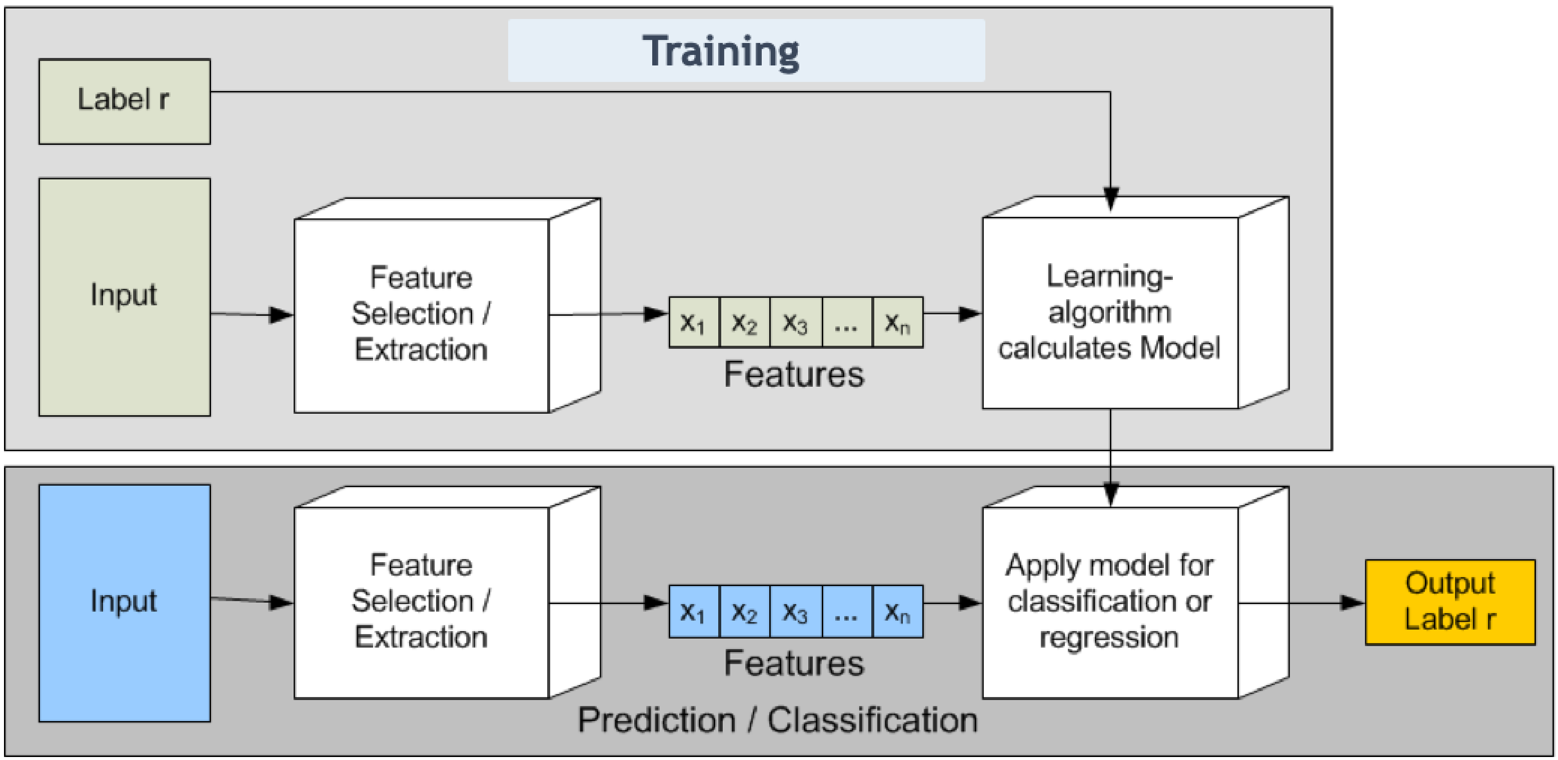
Machine Learning and LLM

Basics

- For supervised text classification, we usually need:
  - **Machine Learning Model**
  - **Labeled Data**
    - *Training Data*
    - *Test / Validation Data*
  - (Feature Extraction Method)



Basic Idea behind Machine Learning

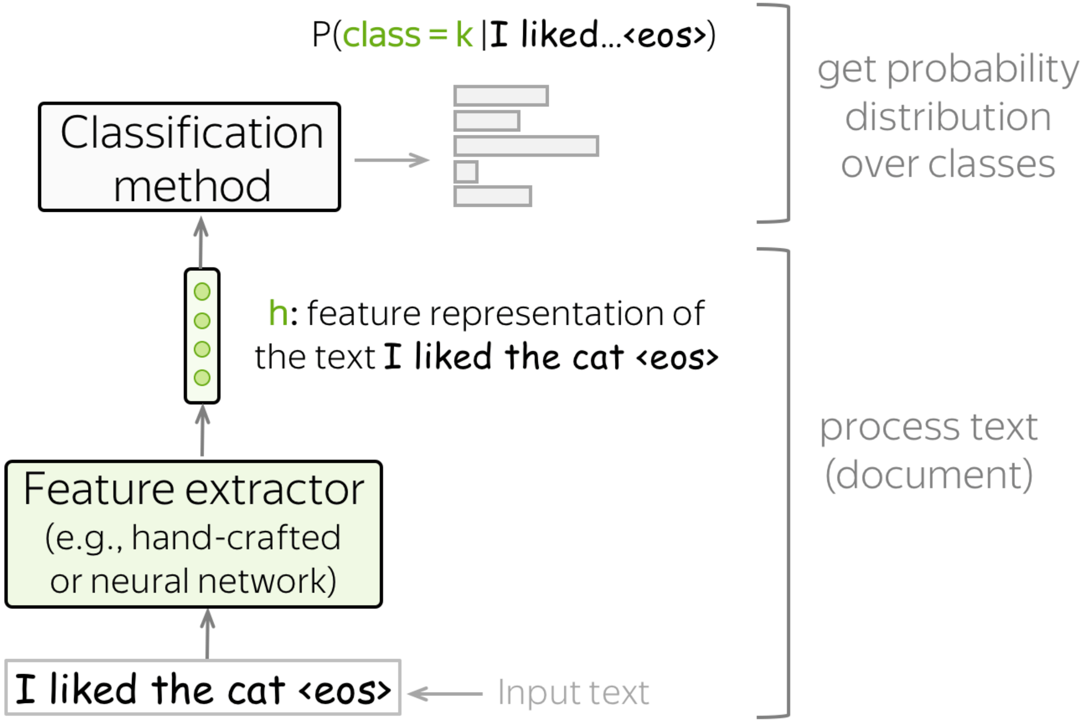


https://griesshaber.pages.mi.hdm-stuttgart.de/nlp/06classification/06classification.html

gesis Leibniz-Institut für Sozialwissenschaften

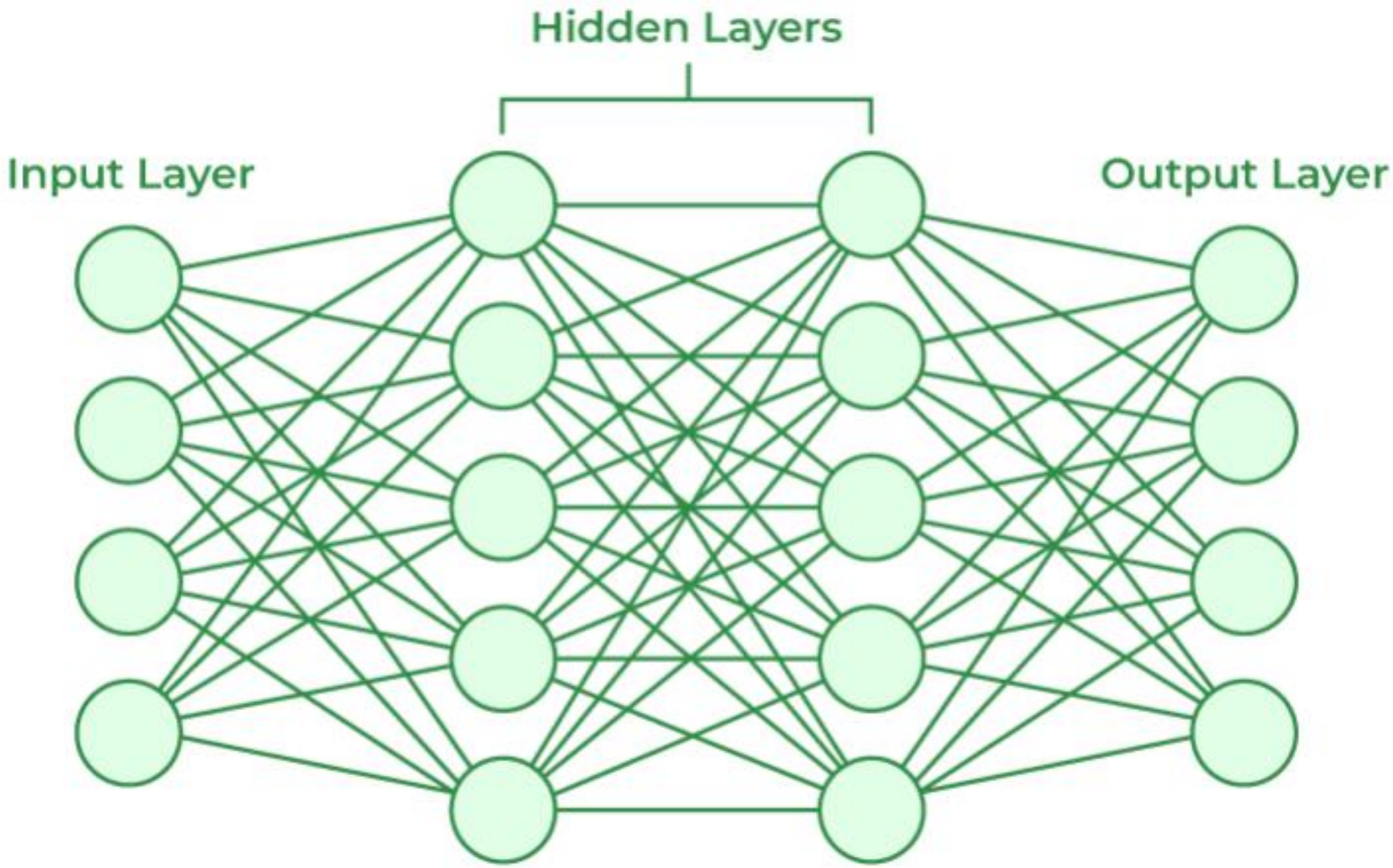
Text classifiers have the following structure

- **Feature extractor**
  - Makes the text machine-readable
  - Either manually defined or learned (e.g., with neural networks)
  - **Same** for **multi-class** and **multi-label** classification
- **Classifier**
  - Assigns class probabilities given feature representation of a text
  - **Different** for **multi-class** and **multi-label** classification



https://lena-voita.github.io/nlp_course/text_classification.html#general_view

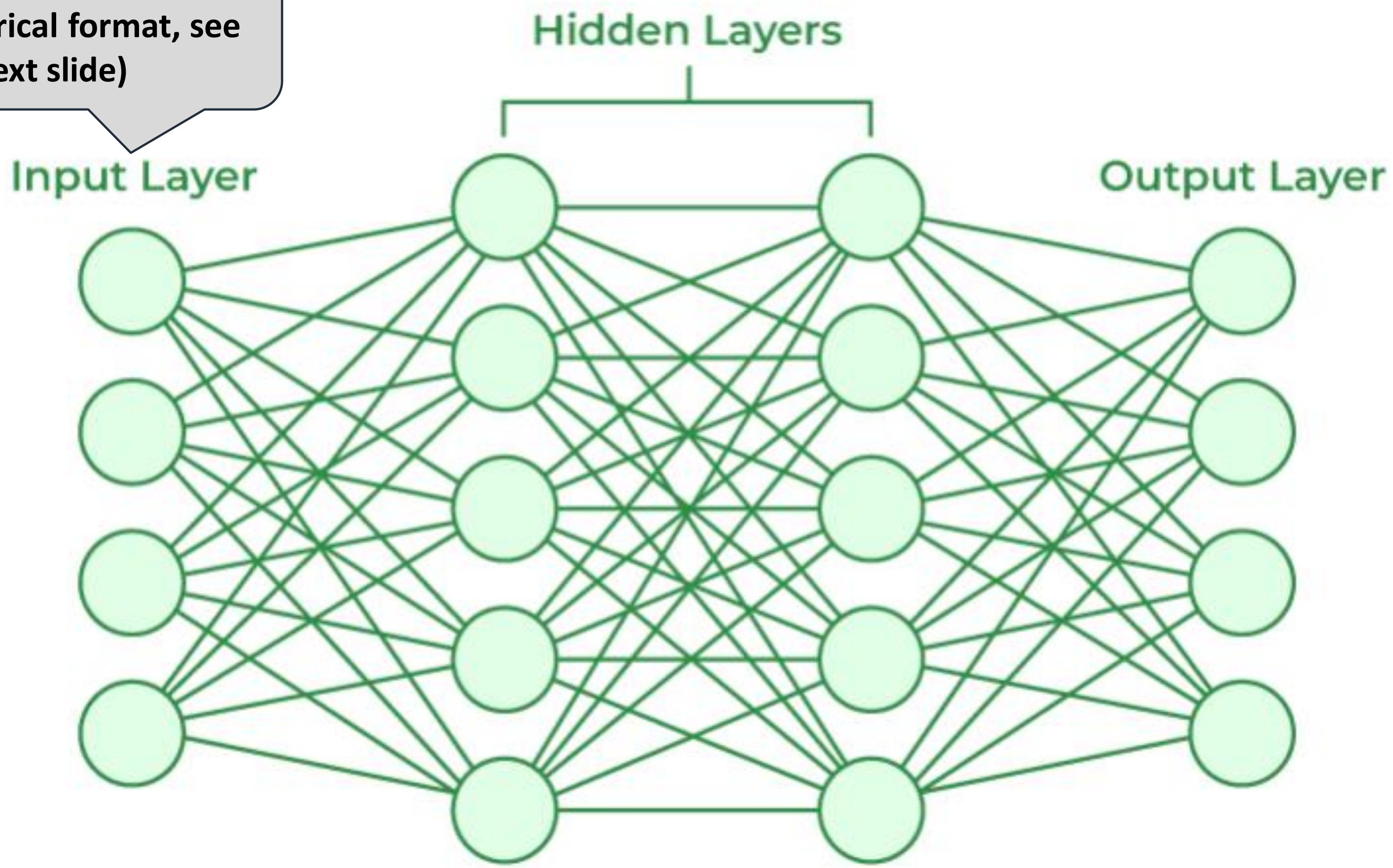
Neural Networks



https://medium.com/@asifurrahmanaust/lesson-3-neural-network-is-nothing-but-a-linear-regression-e05a328a0f23

Neural Networks

Input of the text
(in numerical format, see next slide)

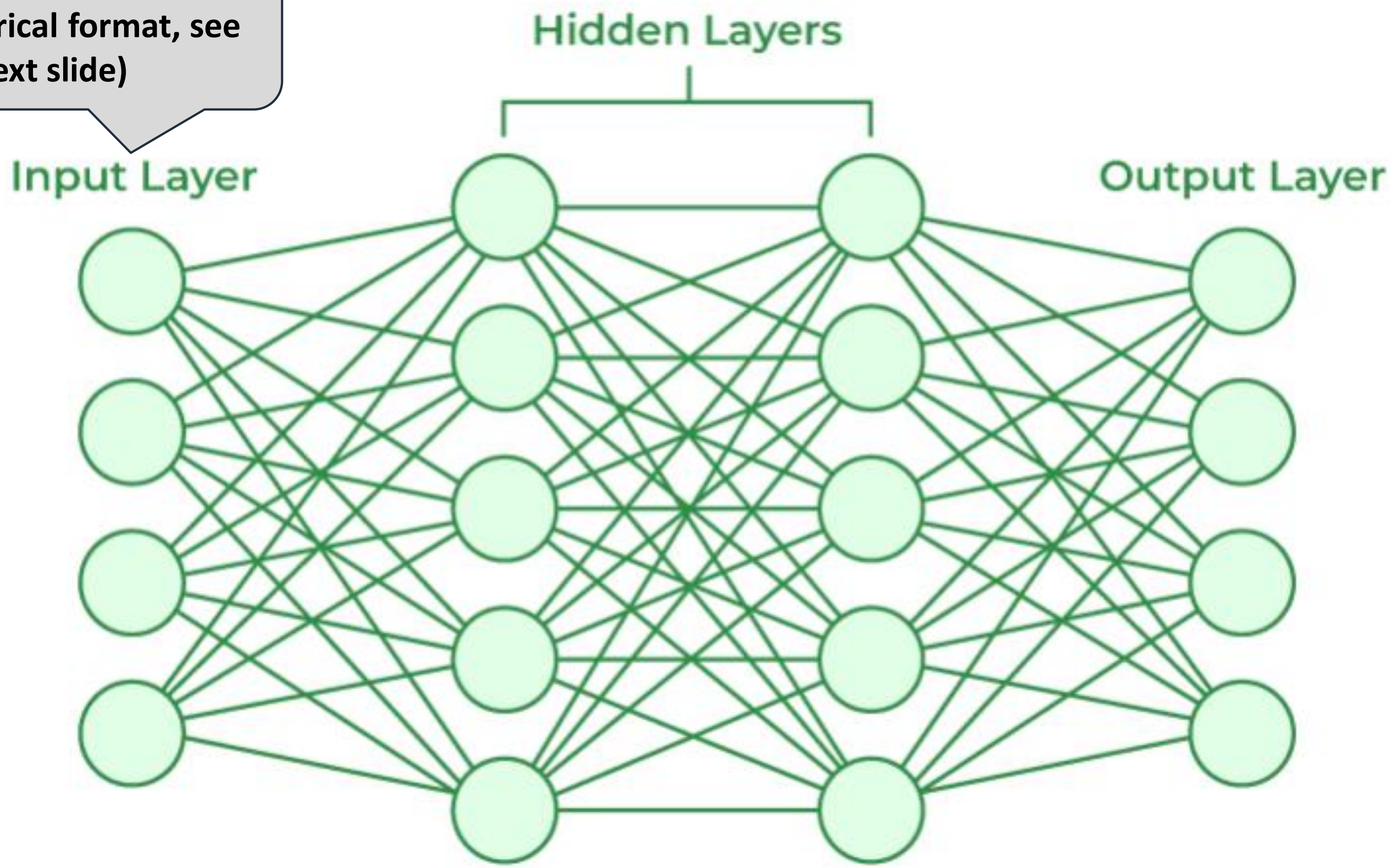


https://medium.com/@asifurrahmanaust/lesson-3-neural-network-is-nothing-but-a-linear-regression-e05a328a0f23

Neural Networks

Network of neurons with adjustable weights and biases that learn from data during training

Input of the text (in numerical format, see next slide)



https://medium.com/@asifurrahmanaust/lesson-3-neural-network-is-nothing-but-a-linear-regression-e05a328a0f23

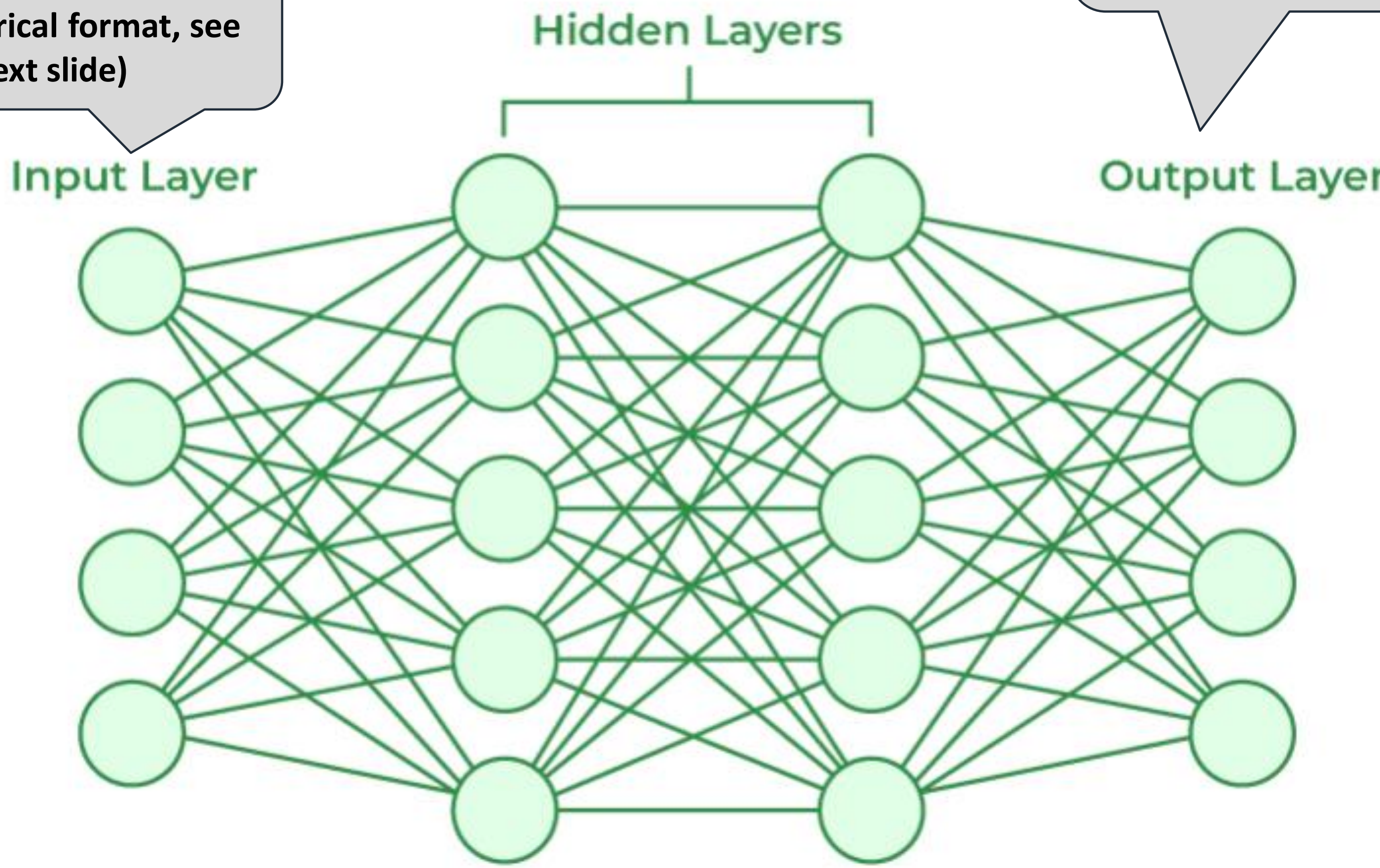
gesis Leibniz-Institut für Sozialwissenschaften

Neural Networks

Network of neurons with adjustable weights and biases that learn from data during training

The number of output scores ("Logits") is equal to the number of labels m

Input of the text (in numerical format, see next slide)

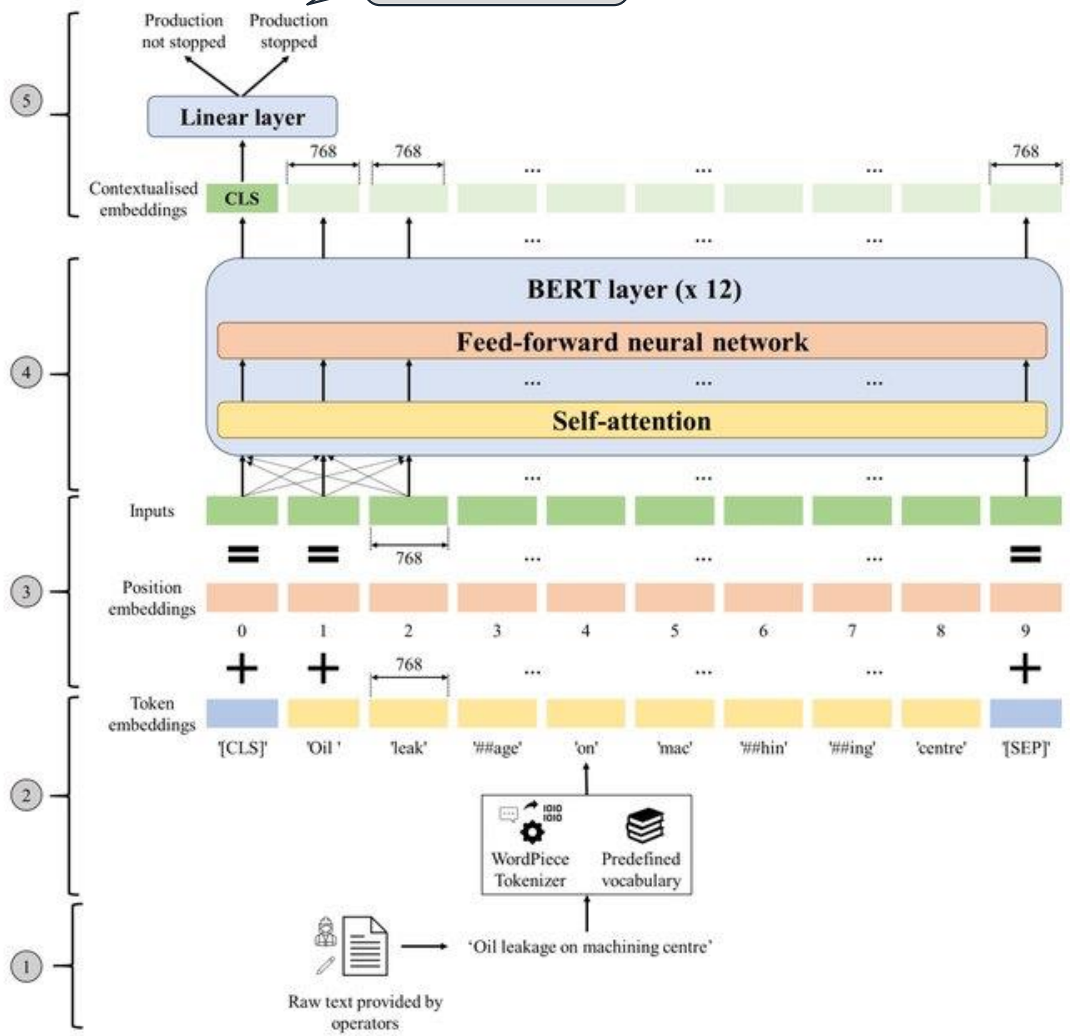


https://medium.com/@asifurrahmanaust/lesson-3-neural-network-is-nothing-but-a-linear-regression-e05a328a0f23

gesis Leibniz-Institut für Sozialwissenschaften

Transformer

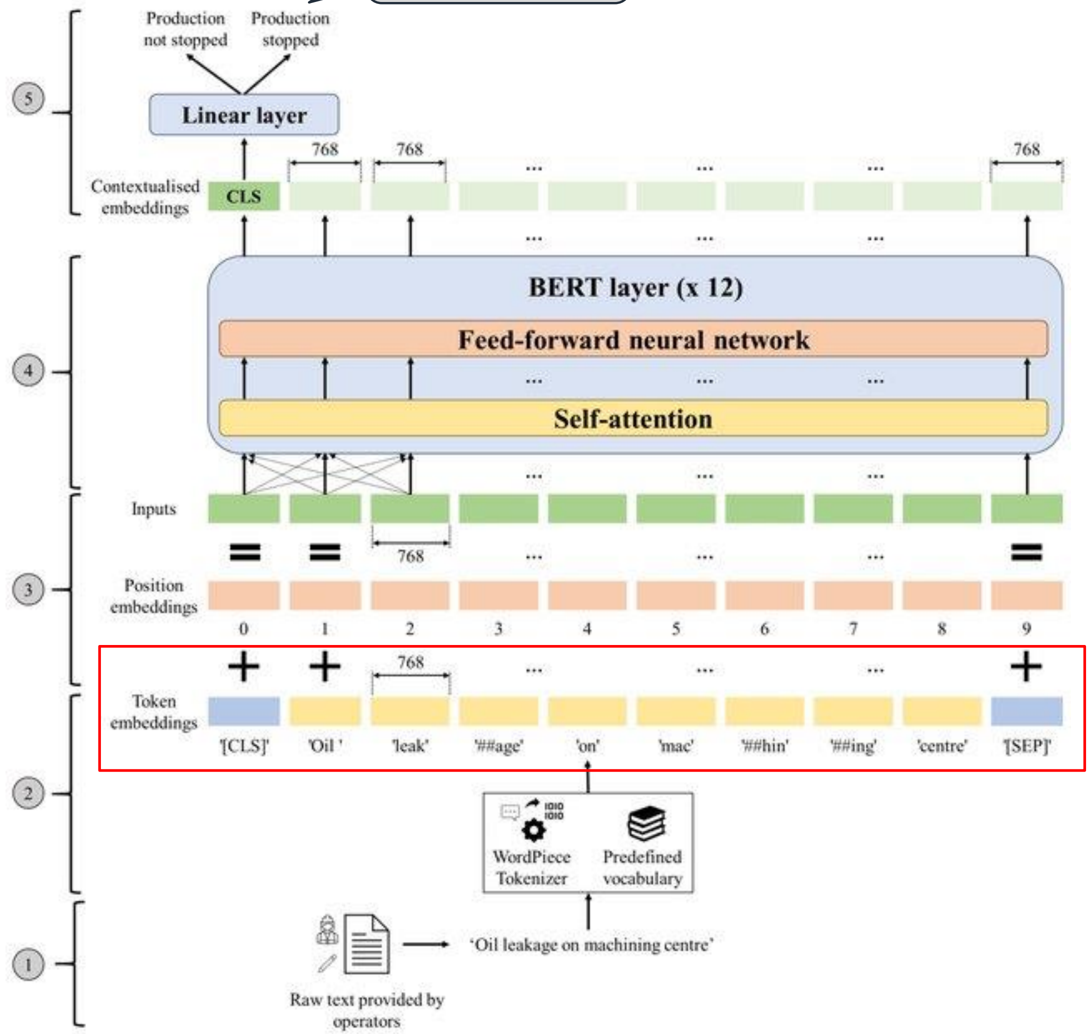
m_{Labels} = 2



https://www.researchgate.net/publication/353419108_Using_deep_learning_to_value_free-form_text_data_for_predictive_maintenance/figures?lo=1

Transformer

m_{Labels} = 2



Token Embeddings

https://www.researchgate.net/publication/353419108_Using_deep_learning_to_value_free-form_text_data_for_predictive_maintenance/figures?lo=1

Embeddings

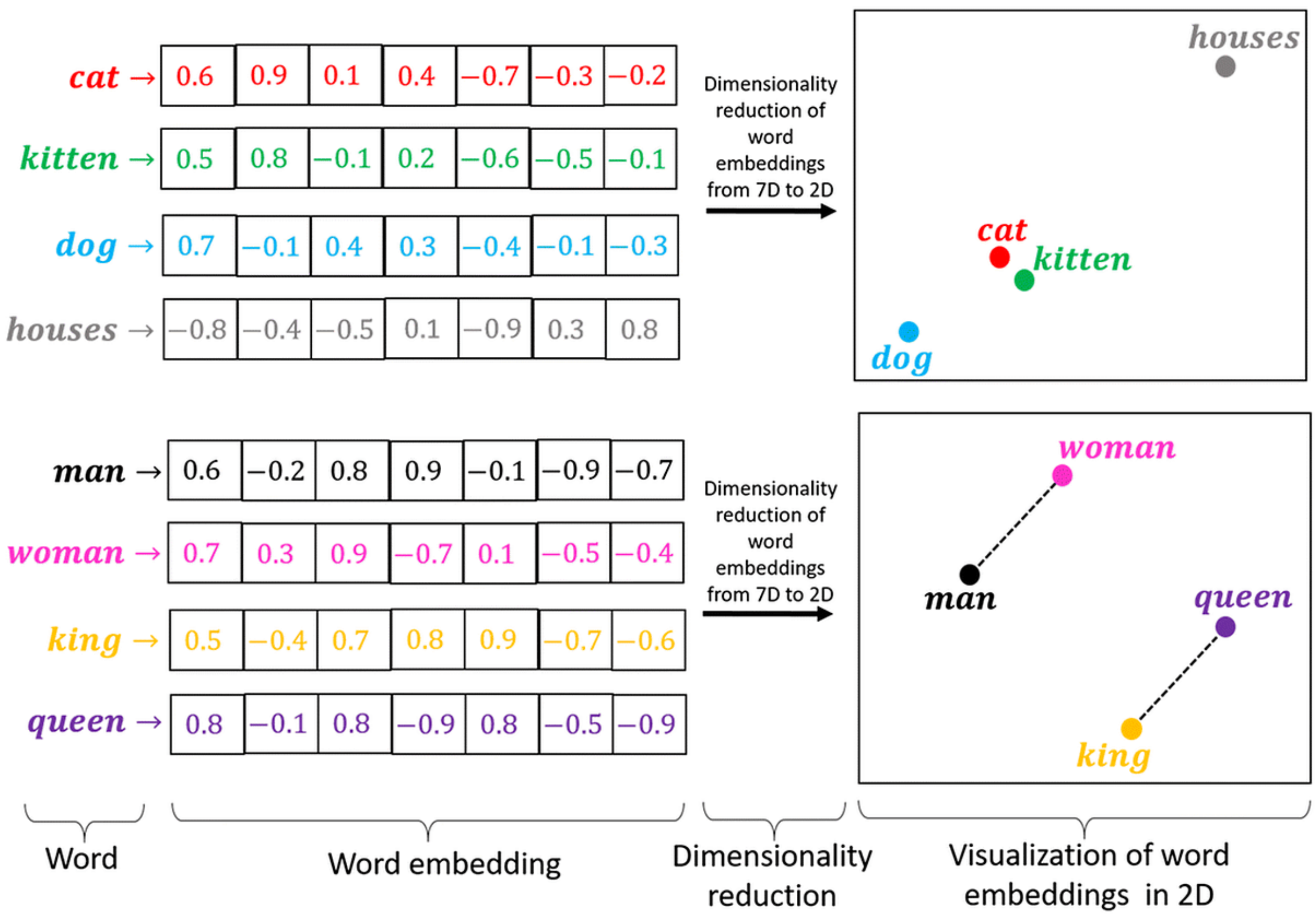
```
Token String    Token ID        Embedded Token Vector
     '<s>' ->    0 -> [ 0.1150, -0.1438,  0.0555, ... ]
   '<pad>' ->    1 -> [ 0.1149, -0.1438,  0.0547, ... ]
    '</s>' ->    2 -> [ 0.0010, -0.0922,  0.1025, ... ]
   '<unk>' ->    3 -> [ 0.1149, -0.1439,  0.0548, ... ]
       '.' ->    4 -> [-0.0651, -0.0622, -0.0002, ... ]
    ' the' ->    5 -> [-0.0340,  0.0068, -0.0844, ... ]
       ',' ->    6 -> [ 0.0483, -0.0214, -0.0927, ... ]
     ' to' ->    7 -> [-0.0439,  0.0201,  0.0189, ... ]
    ' and' ->    8 -> [ 0.0523, -0.0208, -0.0254, ... ]
     ' of' ->    9 -> [-0.0732,  0.0070, -0.0286, ... ]
      ' a' ->   10 -> [-0.0194,  0.0302, -0.0838, ... ]
                                ...
```

https://www.lesswrong.com/posts/pHPmMGEMYefk9jLeh/llm-basics-embedding-spaces-transformer-token-vectors-are

Embeddings

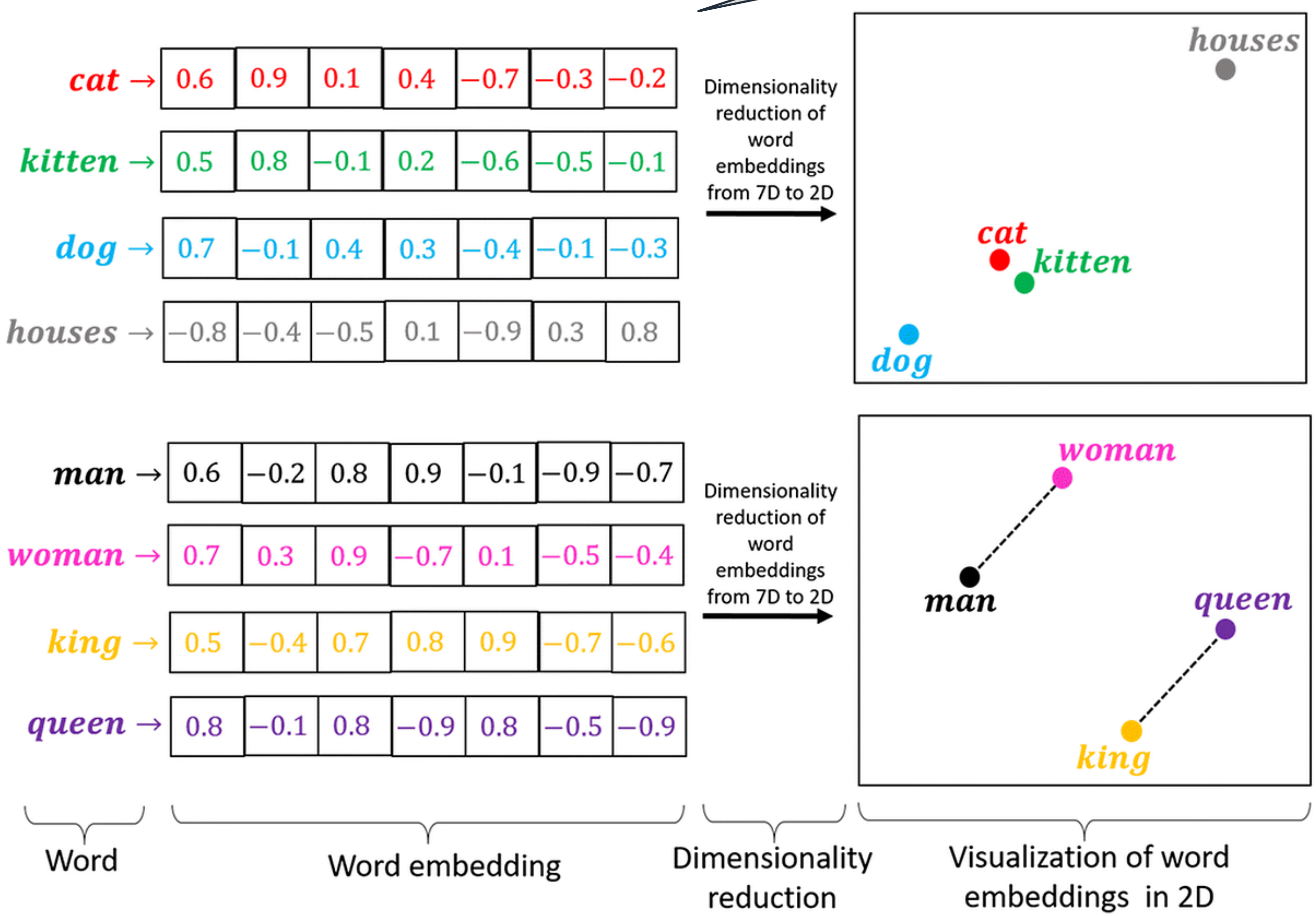
- Input of modern machine learning models (LLMs)
- Basic idea: Words which frequently appear in similar contexts have similar meaning (distributional hypotheses)
- **Embeddings**:
  - **Numerical representations** of words/token in high-dimensional vectors
  - Vectors have fixed length
  - **Information** about words **semantic properties and syntactic functions** are distributed across dimensions
  - Words that are **close to each other semantically** (e.g., “cat” and “kitten”) **are close in the vector space**

Feature Extraction: Word-embeddings



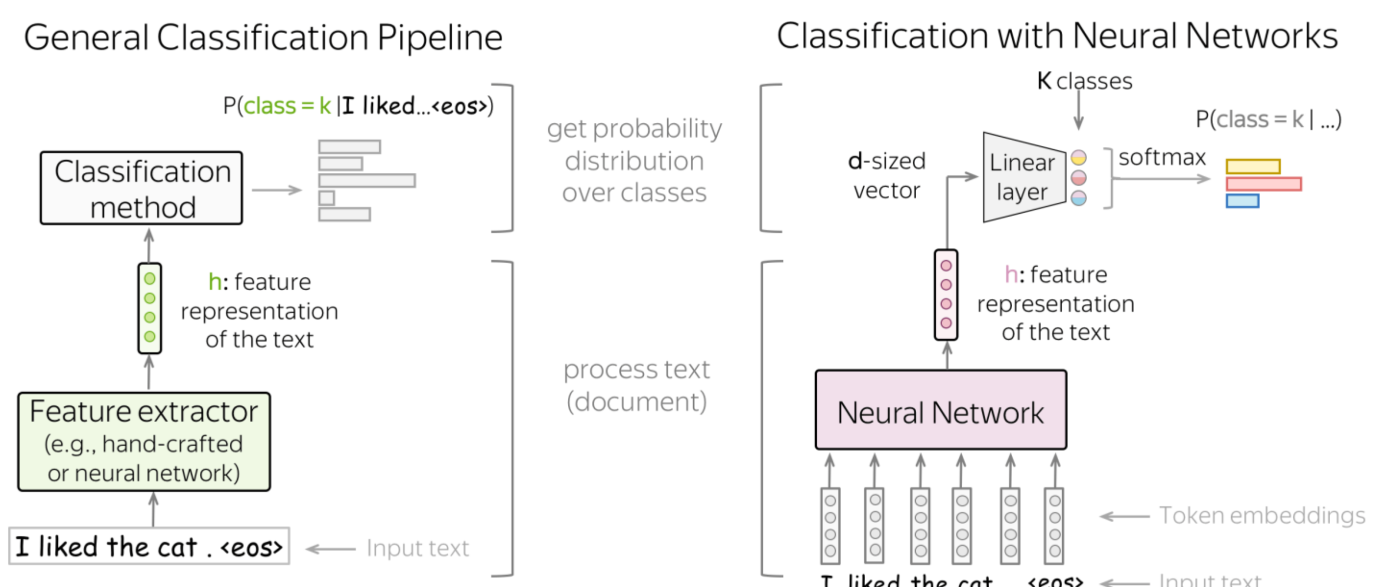
Feature Extraction: Word-embeddings

Usually, we do not know what the dimensions stand for (n_{dim}>700)



Feature Extraction: Word-embeddings

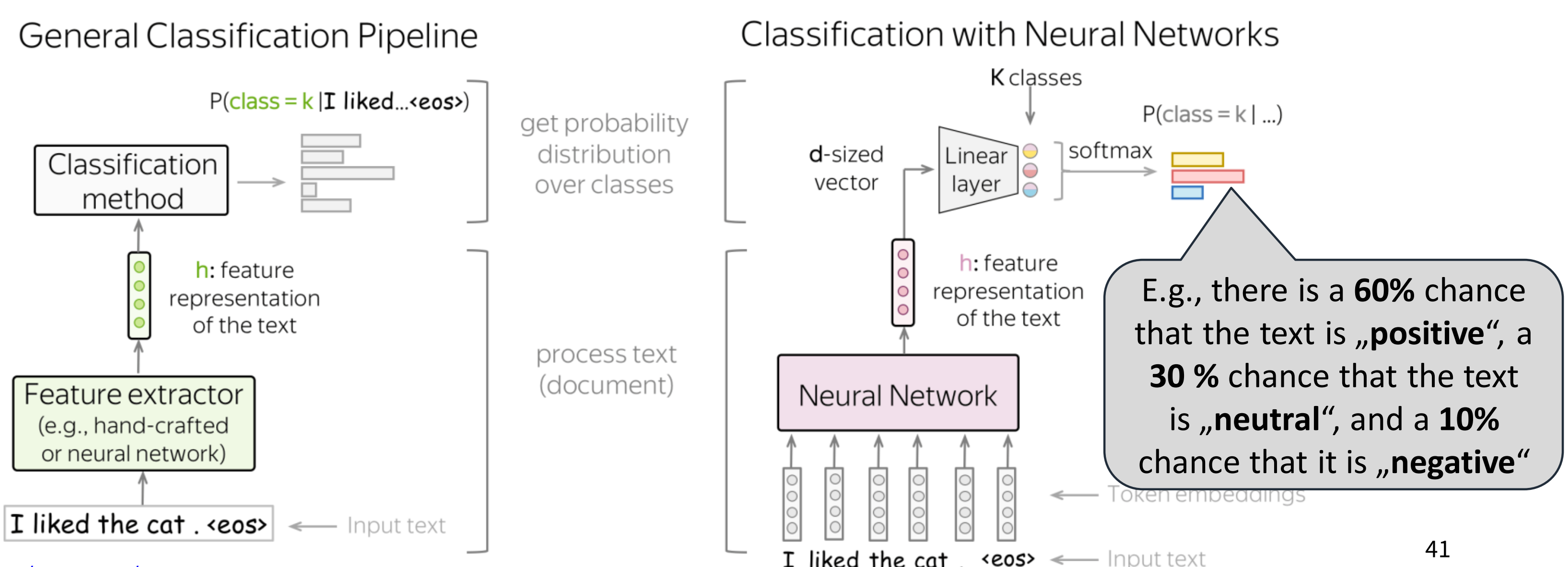
- For feature extraction, we feed the embeddings of the input tokens to a neural network
- The neural network gives us a vector representation of the input text
- Ultimately, this vector is used for classification.



https://lena-voita.github.io/nlp_course/text_classification.html#multi_label

Feature Extraction: Word-embeddings

- For feature extraction, we feed the embeddings of the input tokens to a neural network
- The neural network gives us a vector representation of the input text
- Ultimately, this vector is used for classification.



https://lena-voita.github.io/nlp_course/text_classification.html#multi_label

Validation

Validation

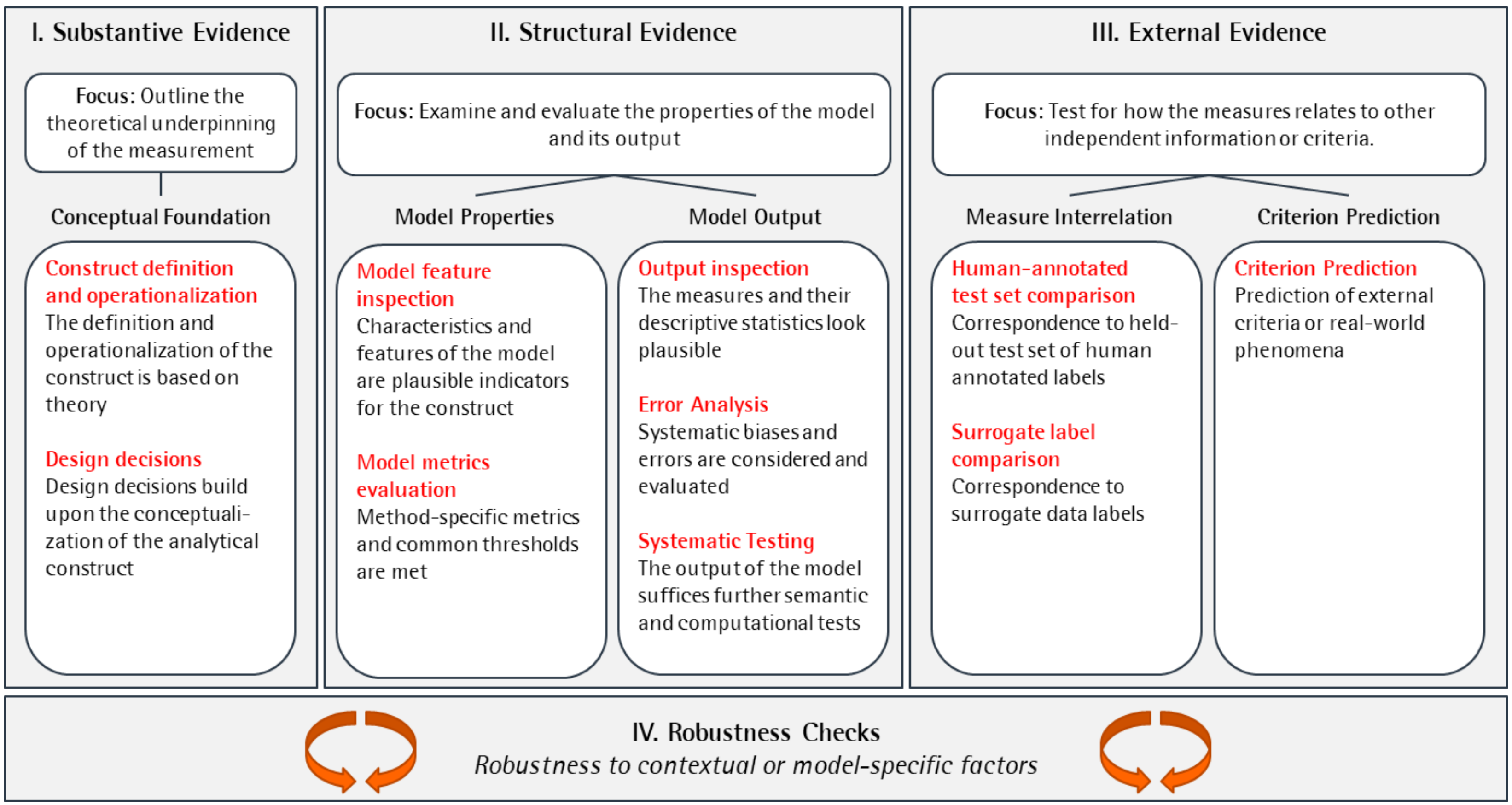
- Validation is critical task for any classification task
- Making sure that the classification has
  - Little error (random)
  - Free from Bias (systematic)
- Two broad categories
  - **Internal Validation** (i.e., evaluating the measures and model features, error analysis etc.)
  - **External Validation** (i.e., comparing with gold-standard data)
- Especially for multi-dimensional social science constructs, validation should be taken seriously!

https://www.tandfonline.com/doi/full/10.1080/19312458.2023.2285765

gesis Leibniz-Institut für Sozialwissenschaften

Validation

For more information: https://www.tandfonline.com/doi/full/10.1080/19312458.2023.2285765



https://www.valitex.info/
https://arxiv.org/abs/2307.02863

Validation

I. Substantive Evidence

Focus: Outline the theoretical underpinning of the measurement

Conceptual Foundation

Construct definition and operationalization
The definition and operationalization of the construct is based on theory

Design decisions
Design decisions build upon the conceptualization of the analytical construct

II. Structural Evidence

Focus: Examine and evaluate the properties of the model and its output

Model Properties

Model feature inspection
Characteristics and features of the model are plausible indicators for the construct

Model metrics evaluation
Method-specific metrics and common thresholds are met

Model Output

Output inspection
The measures and their descriptive statistics look plausible

Error Analysis
Systematic biases and errors are considered and evaluated

Systematic Testing
The output of the model suffices further semantic and computational tests

III. External Evidence

Focus: Test for how the measures relates to other independent information or criteria.

Measure Interrelation

Human-annotated test set comparison
Correspondence to held-out test set of human annotated labels

Surrogate label comparison
Correspondence to surrogate data labels

Criterion Prediction

Criterion Prediction
Prediction of external criteria or real-world phenomena

IV. Robustness Checks

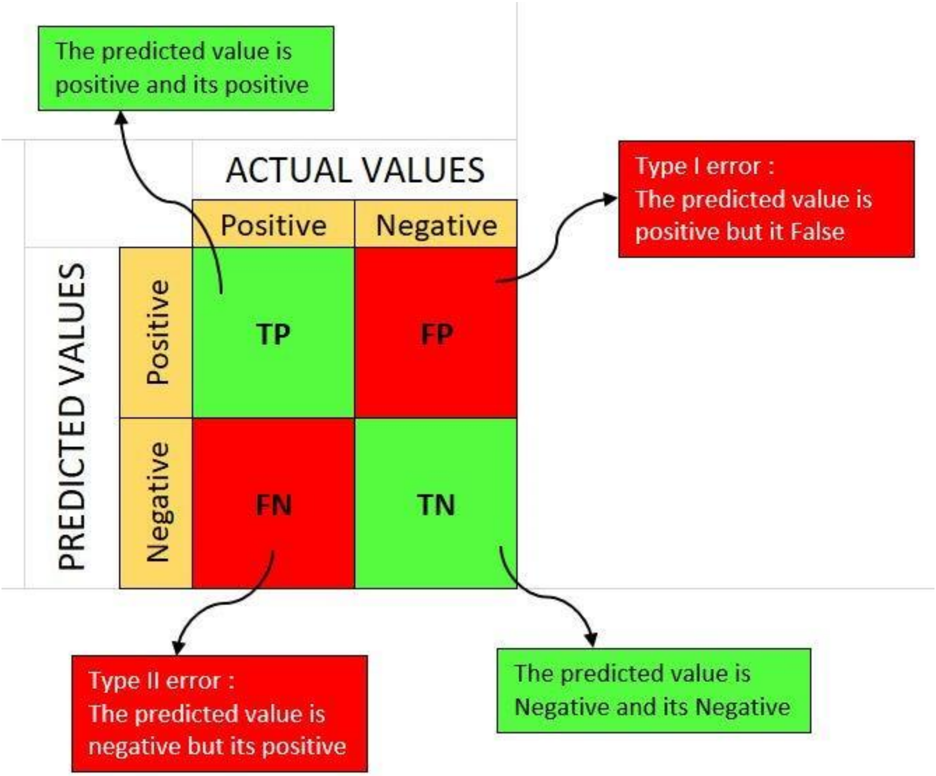
Robustness to contextual or model-specific factors

https://www.valitex.info/
https://arxiv.org/abs/2307.02863

Validation: Comparison with Labels

- Use Case: We want to classify a text as either having “positive” or “negative” tone

- We therefore compare our predictions against some “actual values” and document our guesses in a **Confusion Matrix**



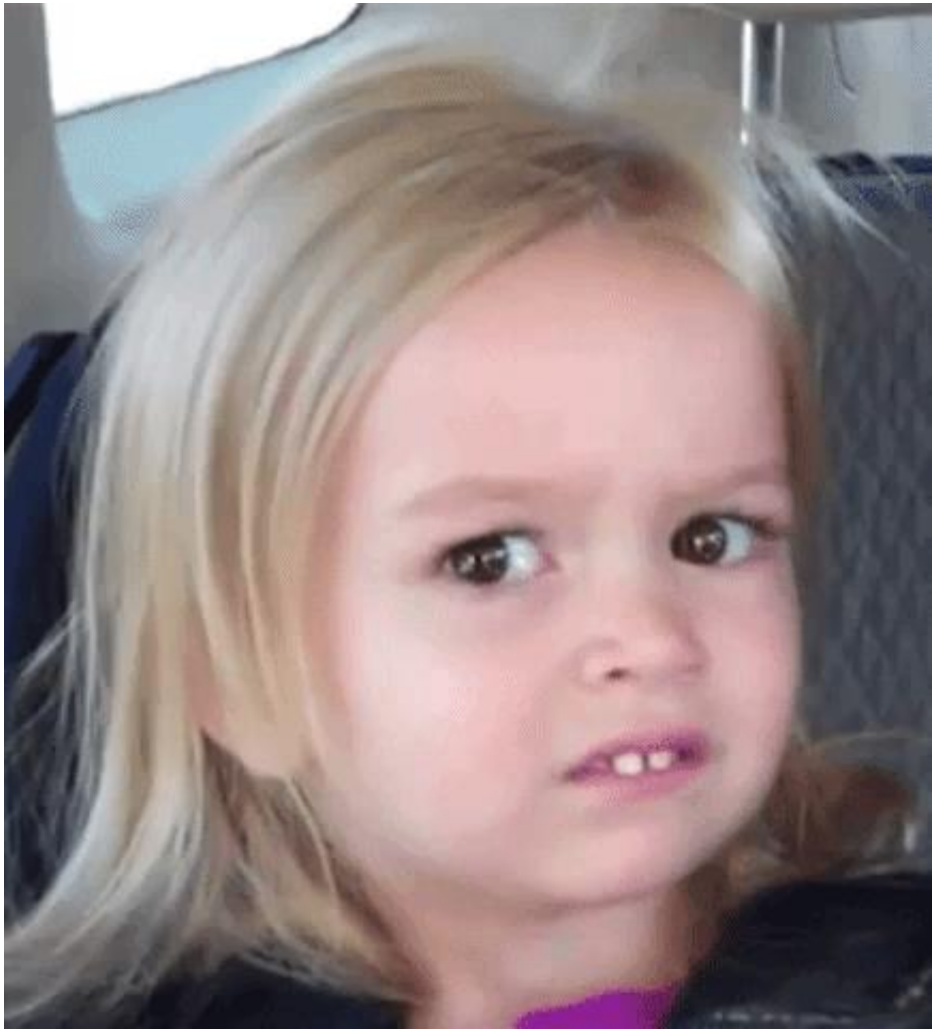
https://medium.com/analytics-vidhya/what-is-a-confusion-matrix-d1c0f8feda5

Evaluation Metrics

$$
\begin{aligned}
precision &= \frac{TP}{TP + FP} \\
recall &= \frac{TP}{TP + FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall} \\
accuracy &= \frac{TP + TN}{TP + FN + TN + FP} \\
specificity &= \frac{TN}{TN + FP}
\end{aligned}
$$

Evaluation Metrics

$$\begin{aligned}
precision &= \frac{TP}{TP+FP} \\
recall &= \frac{TP}{TP+FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall} \\
accuracy &= \frac{TP+TN}{TP+FN+TN+FP} \\
specificity &= \frac{TN}{TN+FP}
\end{aligned}$$



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

Accuracy

- We can first calculate the overall ***accuracy*** of our classifier

Accuracy = TP + TN / TP + TN + FP + FN
= 30+56/30+56+12+8
= **0.81**

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

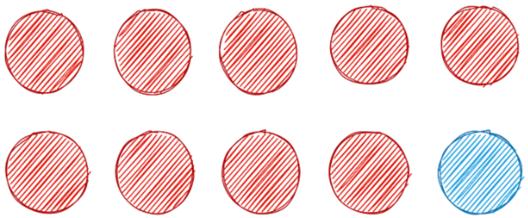
		Predicted	
		0	**1**
Actual	**0**	30	12
	1	8	56

https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall

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

Why we need different metrics

- E.g., for imbalanced data, accuracy does not give the full picture
- When everything is classified as red, our classifier would have an accuracy of 90%
  - True positive = 0 (we never predict the positive class)
  - True negative = 9 (we always predict the negative class)
  - False positive = 0 (we never predict the positive class)
  - False Negative = 1 (we labeled the positive class as neg)



- But is this a good classifier?
  What about the predictions on the blue classes?

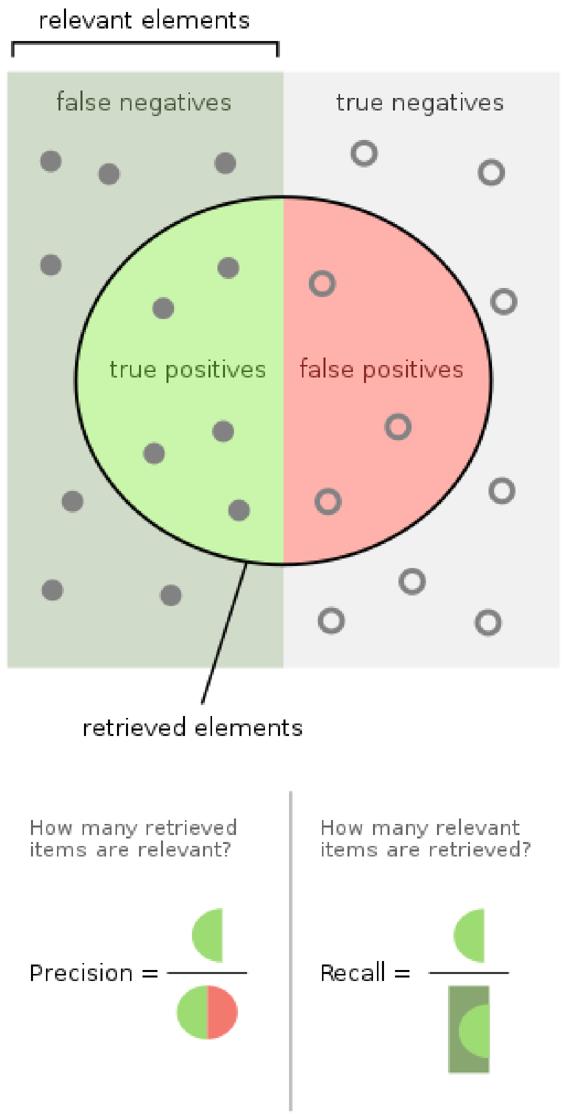
https://towardsdatascience.com/evaluating-multi-label-classifiers-a31be83da6ea

gesis Leibniz-Institut für Sozialwissenschaften

Precision and Recall

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

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



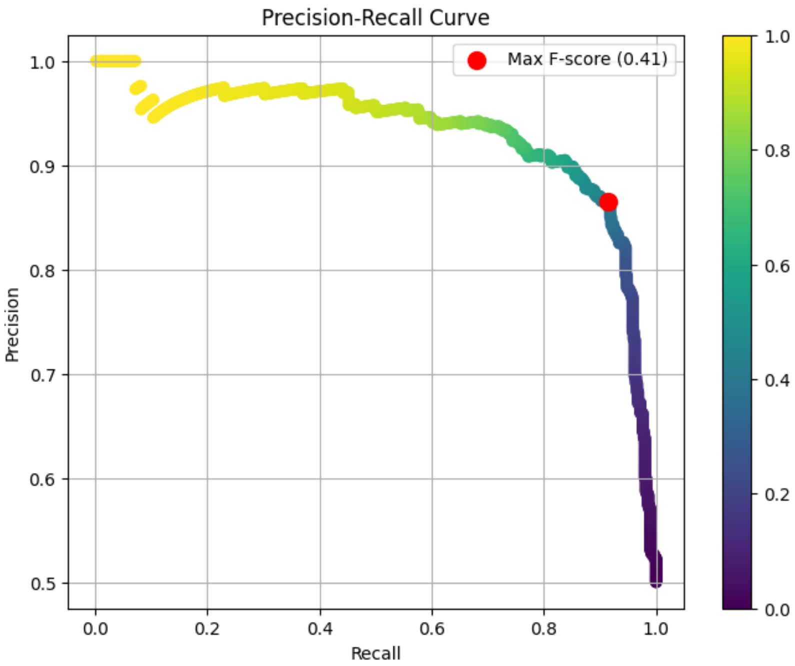
- **Precision (Positive Predictive Value)**
  - Definition: The ratio of correctly predicted positive observations to the total predicted positive observations.
  - Importance: Critical in scenarios where the cost of false positives is high (e.g., pregnancy test)
- **Recall (Sensitivity, True Positive Rate)**
  - Definition: The ratio of correctly predicted positive observations to all observations that are positive
  - Importance: Essential in situations where missing a positive case has a significant consequence (e.g., COVID-test at the beginning of the pandemic)

F1 Score

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

- The F1 score is the harmonic mean of precision and recall
- **It thus symmetrically represents both precision and recall in one metric**

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$



Take-aways Validation

- Calculating (average) accuracy, precision, recall, and F1-score is possible for both **multi-class** and **multi-label** classification (we will soon see how)
- Provide immediate metrics of model performance
- Software automates calculation
- More validation is required if the quality (truthfulness) of your predictions is important

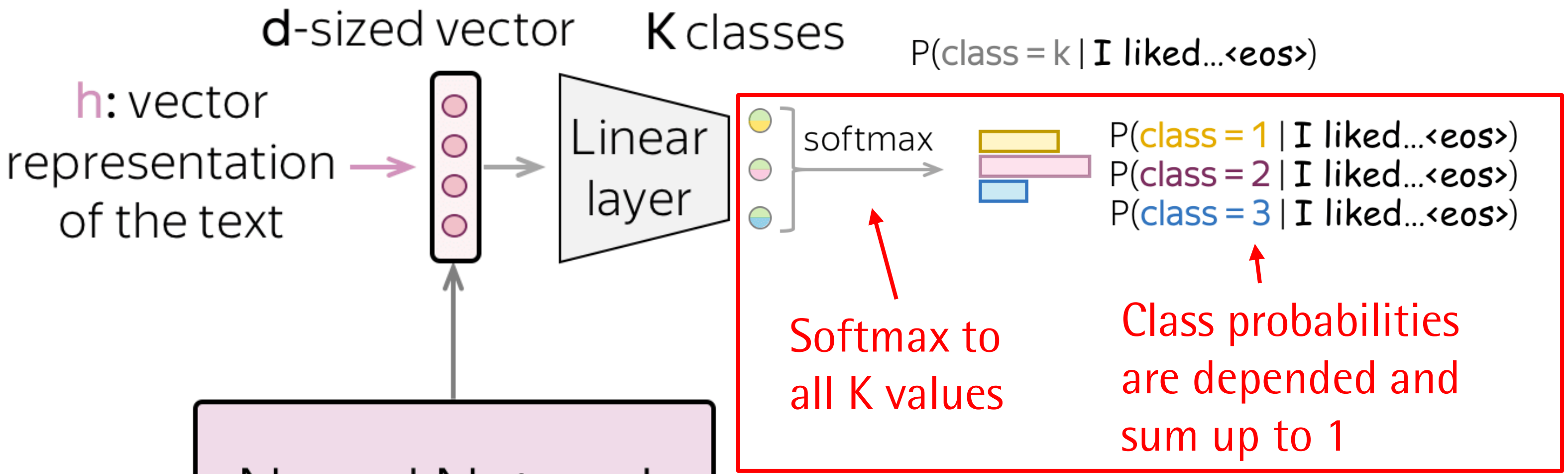
Multi-Class Classification

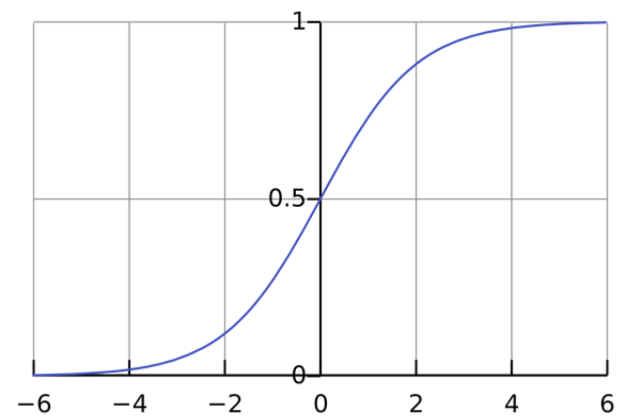
Definitions

- **Multi-class classification** is a classification task with more than two classes where **each sample can only be labeled** as one class.
- Labels are mutually exclusive
- Can be seen as an extension of binary classification
- Requires (usually) no problem transformation
- Probabilities for each class add up to 100%
- E.g., sentiment of a text (positive, neutral, negative)

https://scikit-learn.org/stable/modules/multiclass.html#multilabel-classification

Multi-Class Classification





https://lena-voita.github.io/nlp_course/text_classification.html#multi_label

Validation

Main Take-away:
For both **multi-class** and **multi-label** classification, we need to calculate **average performance metrics**!

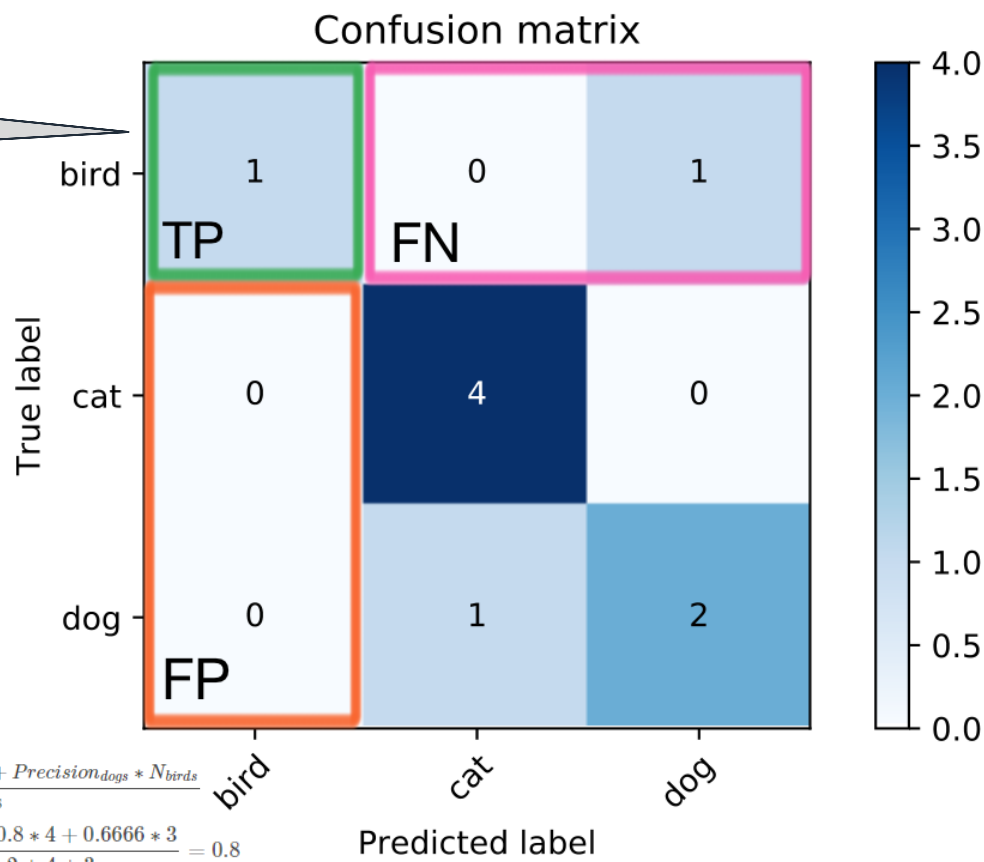
Multi-Class Confusion Matrix

„bird“ is the reference category here

	TP	FP	FN	Precision	Number of samples
bird	1	0	1	1	2
cat	4	1	0	0.8	4
dog	2	1	1	0.667	3
TOTAL	7	2	2		

$$\text{Micro-averaged Precision} = \frac{TP_{total}}{TP_{total} + FP_{total}} = \frac{7}{7+2} = 0.7777$$

$$\text{Macro-averaged Precision} = \frac{1}{3} Precision_{birds} + Precision_{cats} + Precision_{dogs}$$
$$= \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$



$$\text{Weighted-averaged Precision} = \frac{Precision_{birds} * N_{birds} + Precision_{cats} * N_{birds} + Precision_{dogs} * N_{birds}}{\text{Total number of samples}}$$
$$= \frac{1*2 + 0.8*4 + 0.6666*3}{2+4+3} = 0.8$$

https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/

Multi-Class Confusion Matrix

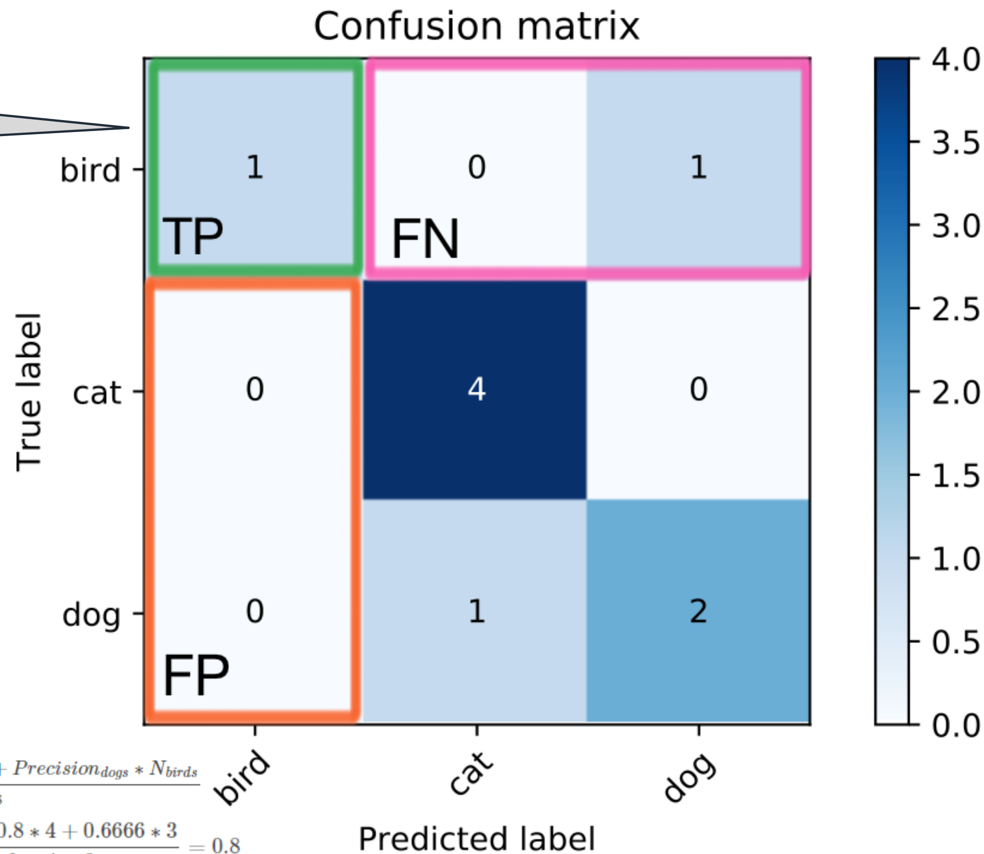
„bird" is the reference category here

	TP	FP	FN	Precision	Number of samples
bird	1	0	1	1	2
cat	4	1	0	0.8	4
dog	2	1	1	0.667	3
TOTAL	7	2	2		

$$\text{Micro-averaged Precision} = \frac{TP_{total}}{TP_{total} + FP_{total}} = \frac{7}{7+2} = 0.7777$$

$$\text{Macro-averaged Precision} = \frac{1}{3} Precision_{birds} + Precision_{cats} + Precision_{dogs}$$
$$= \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$

$$\text{Weighted-averaged Precision} = \frac{Precision_{birds} * N_{birds} + Precision_{cats} * N_{birds} + Precision_{dogs} * N_{birds}}{\text{Total number of samples}}$$
$$= \frac{1 * 2 + 0.8 * 4 + 0.6666 * 3}{2 + 4 + 3} = 0.8$$



https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/

gesis Leibniz-Institut für Sozialwissenschaften

Multi-Class Confusion Matrix

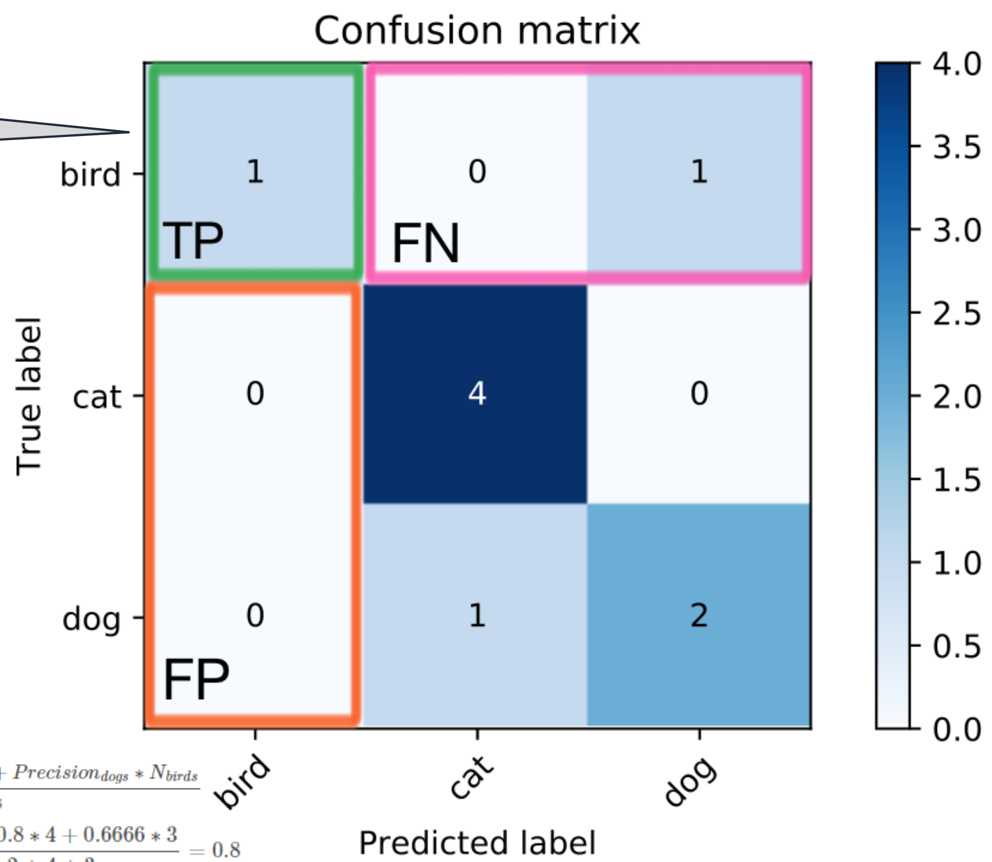
„bird“ is the reference category here

	TP	FP	FN	Precision	Number of samples
bird	1	0	1	1	2
cat	4	1	0	0.8	4
dog	2	1	1	0.667	3
TOTAL	7	2	2		

$$\text{Micro-averaged Precision} = \frac{TP_{total}}{TP_{total} + FP_{total}} = \frac{7}{7+2} = 0.7777$$

$$\text{Macro-averaged Precision} = \frac{1}{3} Precision_{birds} + Precision_{cats} + Precision_{dogs}$$
$$= \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$

$$\text{Weighted-averaged Precision} = \frac{Precision_{birds} * N_{birds} + Precision_{cats} * N_{birds} + Precision_{dogs} * N_{birds}}{\text{Total number of samples}}$$
$$= \frac{1 * 2 + 0.8 * 4 + 0.6666 * 3}{2 + 4 + 3} = 0.8$$



Both for **multi-class** and **multi-label** classification, we need to average the metrics across classes

	TP	FP	FN	Precision	Number of samples
bird	1	0	1	1	2
cat	4	1	0	0.8	4
dog	2	1	1	0.667	3
TOTAL	7	2	2		

- **Micro-averaged**: all samples equally contribute to the final averaged metric

$$\text{Micro-averaged Precision} = \frac{TP_{total}}{TP_{total} + FP_{total}} = \frac{7}{7+2} = 0.7777$$

- **Macro-averaged**: all classes equally contribute to the final averaged metric

$$\text{Macro-averaged Precision} = \frac{1}{3} Precision_{birds} + Precision_{cats} + Precision_{dogs}$$
$$= \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$

- **Weighted-averaged**: each classes's contribution to the average is weighted by its size

$$\text{Weighted-averaged Precision} = \frac{Precision_{birds} * N_{birds} + Precision_{cats} * N_{birds} + Precision_{dogs} * N_{birds}}{\text{Total number of samples}}$$
$$= \frac{1*2 + 0.8*4 + 0.6666*3}{2+4+3} = 0.8$$

https://iamirmasoud.com/2022/06/19/understanding-micro-macro-and-weighted-averages-for-scikit-learn-metrics-in-multi-class-classification-with-example/

gesis Leibniz-Institut für Sozialwissenschaften

Tutorials

- **Multi-class** classification
  - https://colab.research.google.com/github/lukasbirki/Workshop-Classification/blob/main/Multi-Class%20Classification.ipynb

Multi-Label Classification

gesis Leibniz-Institut für Sozialwissenschaften

Definitions

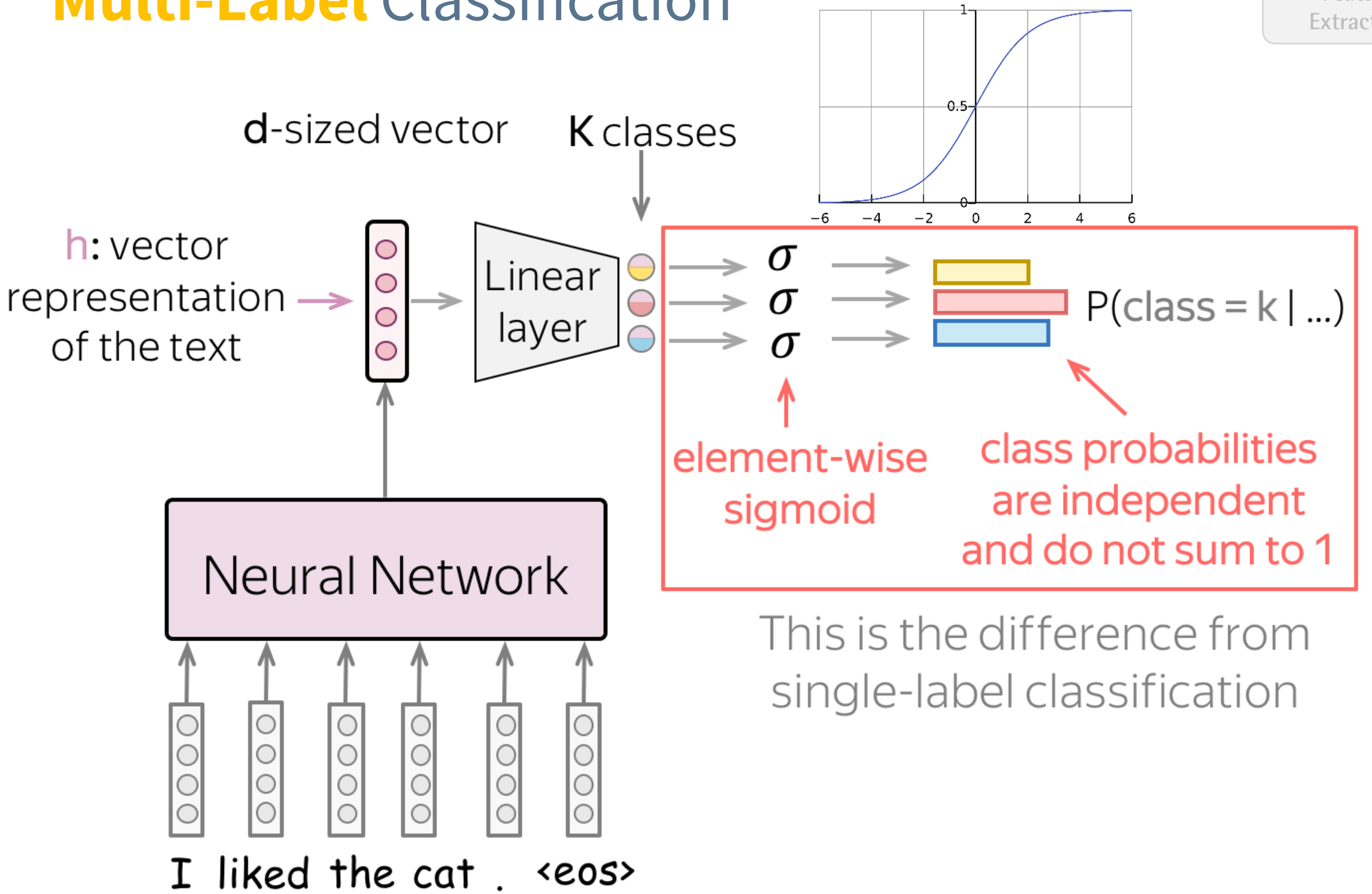
- **Multi-label** classification
  is a classification task labeling each sample with $\boldsymbol{m}$ labels from $\boldsymbol{n_{classes}}$, with $0 \leq \mathbf{m} \leq \mathbf{n_{classes}}$
- Labels are not mutually exclusive
- Can be seen as an extension of multi-class classification
- Can require problem transformation
- Separate probabilities for each output class
- E.g., mentions of characters in a specific book chapter

https://scikit-learn.org/stable/modules/multiclass.html#multilabel-classification

Classification Method

Feature Extraction

Multi-Label Classification



This is the difference from single-label classification

https://lena-voita.github.io/nlp_course/text_classification.html#multi_label

gesis Leibniz-Institut für Sozialwissenschaften

Multi-Label Confusion Matrix

```
expected    predicted
A, C        A, B
C           C
A, B, C     B, C
```

```
expected    predicted
1 0 1       1 1 0
0 0 1       0 0 1
1 1 1       0 1 1
```

$$\begin{bmatrix} TN & FP \\ TP & FN \end{bmatrix}$$

Class A: 1 0
$\boxed{1}$ 1

```
Class B: 1    1
         0    1

Class C: 0    0
         1    2
```

Class A

Precision = TP / (TP + FP)

1 / (1 + 0) = 1

Recall = TP / (TP + FN)

1 / (1 + 1) = 0.5

F1-Score = 0.667

Class B

```
Precision = 0.5
Recall = 1.0
F1-score = 0.667
```

Class C

```
Precision = 1.0
Recall = 0.667
F1-score = 0.8
```

https://towardsdatascience.com/evaluating-multi-label-classifiers-a31be83da6ea

Multi-Label Confusion Matrix

```
expected    predicted
A, C         A, B
C            C
A, B, C      B, C
```

```
expected      predicted
1 0 1         1 1 0
0 0 1         0 0 1
1 1 1         0 1 1
```

$$\begin{bmatrix} TN & FP \\ TP & FN \end{bmatrix}$$

Class A: $\boxed{1}$ 0
1 1

```
Class B: 1   1
         0   1

Class C: 0   0
         1   2
```

Class A

Precision = TP / (TP + FP)
1 / (1 + 0) = 1

Recall = TP / (TP + FN)
1 / (1 + 1) = 0.5

F1-Score = 0.667

Class B

```
Precision = 0.5
Recall = 1.0
F1-score = 0.667
```

Class C

```
Precision = 1.0
Recall = 0.667
F1-score = 0.8
```

https://towardsdatascience.com/evaluating-multi-label-classifiers-a31be83da6ea

gesis Leibniz-Institut für Sozialwissenschaften

Multi-Label Confusion Matrix

```
expected     predicted
A, C          A, B
C             C
A, B, C       B, C
```

```
expected     predicted
1 0 1        1 1 0
0 0 1        0 0 1
1 1 1        0 1 1
```

$$\begin{bmatrix} TN & FP \\ TP & FN \end{bmatrix}$$

Class A: $\begin{matrix} 1 & 0 \\ 1 & \boxed{1} \end{matrix}$

```
Class B: 1   1
         0   1

Class C: 0   0
         1   2
```

Class A

Precision = TP / (TP + FP)

1 / (1 + 0) = 1

Recall = TP / (TP + FN)

1 / (1 + 1) = 0.5

F1-Score = 0.667

Class B

```
Precision = 0.5
Recall = 1.0
F1-score = 0.667
```

Class C

```
Precision = 1.0
Recall = 0.667
F1-score = 0.8
```

https://towardsdatascience.com/evaluating-multi-label-classifiers-a31be83da6ea

Multi-Label Confusion Matrix

```
expected    predicted
A, C         A, B
C            C
A, B, C      B, C
```

```
expected    predicted
1 0 1       1 1 0
0 0 1       0 0 1
1 1 1       0 1 1
```

$$\begin{bmatrix} TN & FP \\ TP & FN \end{bmatrix}$$

Class A: $\boxed{\begin{matrix} 1 & 0 \\ 1 & 1 \end{matrix}}$

```
Class B: 1   1
         0   1

Class C: 0   0
         1   2
```

Class A

Precision = TP / (TP + FP)

1 / (1 + 0) = 1

Recall = TP / (TP + FN)

1 / (1 + 1) = 0.5

F1-Score = 0.667

Class B

```
Precision = 0.5
Recall = 1.0
F1-score = 0.667
```

Class C

```
Precision = 1.0
Recall = 0.667
F1-score = 0.8
```

https://towardsdatascience.com/evaluating-multi-label-classifiers-a31be83da6ea

Repetition

Both for **multi-class** and **multi-label** classification, we need to average the metrics across classes

- **Micro-averaged**: all samples equally contribute to the final averaged metric
- **Macro-averaged**: all classes equally contribute to the final averaged metric
- **Weighted-averaged**: each classes's contribution to the average is weighted by its size

$$\text{Micro-averaged Precision} = \frac{TP_{total}}{TP_{total} + FP_{total}} = \frac{7}{7+2} = 0.7777$$

$$\text{Macro-averaged Precision} = \frac{1}{3} Precision_{birds} + Precision_{cats} + Precision_{dogs}$$
$$= \frac{1}{3}(1 + 0.8 + 0.6666) = 0.8222$$

$$\text{Weighted-averaged Precision} = \frac{Precision_{birds} * N_{birds} + Precision_{cats} * N_{birds} + Precision_{dogs} * N_{birds}}{\text{Total number of samples}}$$
$$= \frac{1 * 2 + 0.8 * 4 + 0.6666 * 3}{2 + 4 + 3} = 0.8$$

https://iamirmasoud.com/2022/06/19/understanding-micro-macro-and-weighted-averages-for-scikit-learn-metrics-in-multi-class-classification-with-example/

gesis Leibniz-Institut für Sozialwissenschaften

Tutorials

- **Multi-label** classification
  - https://colab.research.google.com/github/lukasbirki/Workshop-Classification/blob/main/Multi-Label%20Classification.ipynb

Annotation

Types of Annotation

Annotations are required for

- **Training / Finetuning**
- **Evaluation!**

Multiple ways to annotate text

- **experts**
- **trained coders**
- **crowd workers** (since ~2010)
- **"Zero-Shot Classification of other LLMs/GPT" ?**
- (you already have labelled data, but this is often not the case😭)

- see Krippendorff "Content Analysis: An Introduction to Its Methodology" on experts and trained coders
- see Benoit *et al.* (2016) for optimistic view on crowd coding
  - more opinions: here, here, here
- research on LLMs for annotation: here, here, here, here, and here (but still *many* open meth. questions)

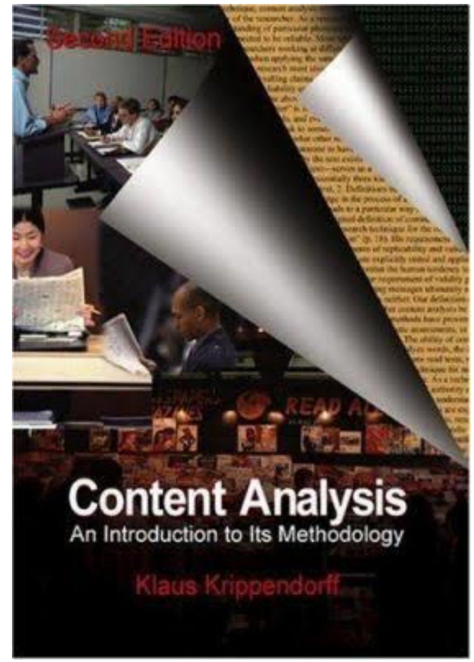
Best Practices

Main Take away:
If humans are unsure how to classify texts, computational methods will fail as well!

Best Practices

- **Concept development**
- **Codebooks & instructions**
- **Coder training**
- **Quality assurance**

Read here for practical guidance



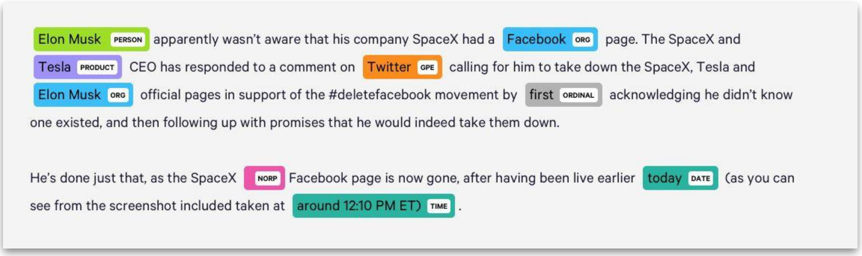
Text data are very context-dependent!
Always inspect your data critically to reflect how your constructs reflect themselves in the text!

gesis Leibniz-Institut für Sozialwissenschaften

Concept development

For instance, concept “populist/non-populist” https://doi.org/10.1017/pan.2022.32

- Level of annotation
  - **document** ⇒ “holistic grading”
  - **paragraph** ⇒ sequence classification (1+ label per para.)
  - **sentence** ⇒ sequence classification (1+ label per sent.)
  - **pairs of sentences** (see here)
  - **word** ⇒ “token classification” (1 label per word, see here)



Quality assurance & assessment

Annotation quality

- important for supervised learning
  - bad annotation result in "noisy" labels
  - noisy labels impair ability to learn the relevant signal
- related to replicability: if coders can agree, task should be replicable
- commonly quantified with inter-coder reliability metrics

Inter-coder reliability

- just % agreement is not enough (need to adjust for baseline)
- compute "chance-adjusted" agreement metrics
  - Krippendorff's alpha
  - Cohen's kappa
- read here and here
- https://github.com/Toloka/crowd-kit

Thank you!

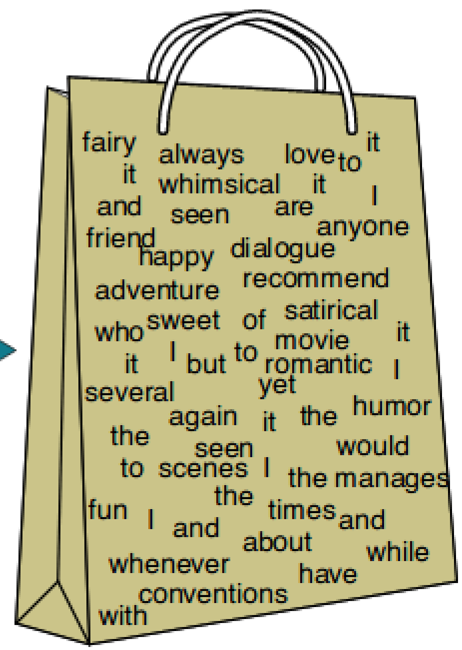
gesis Leibniz-Institut für Sozialwissenschaften

Leibniz Gemeinschaft

Backup

Feature Extraction: Bag of Words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

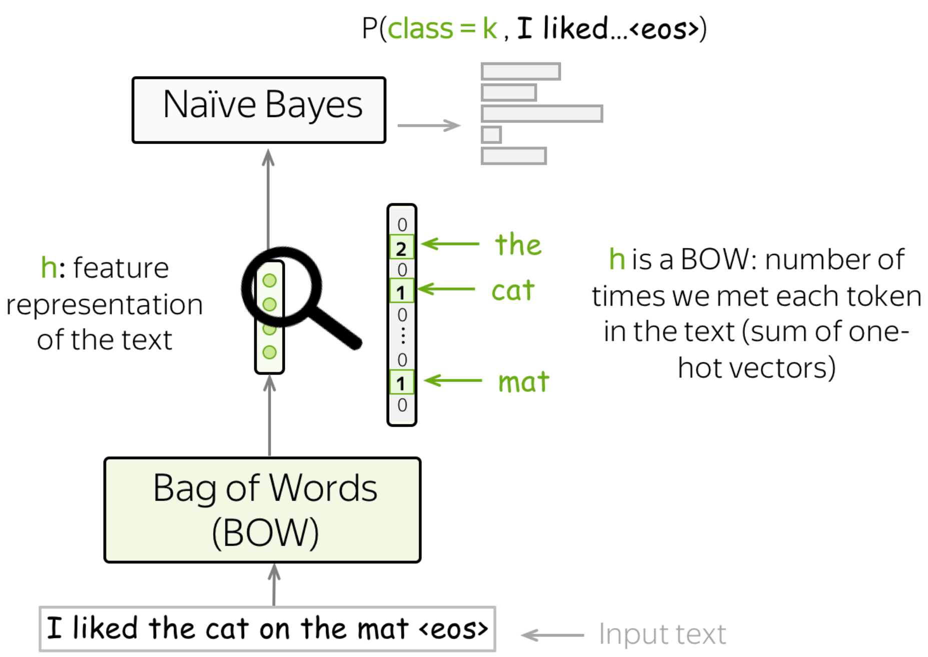


it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

https://dudeperf3ct.github.io/lstm/gru/nlp/2019/01/28/Force-of-LSTM-and-GRU/

Feature Extraction: Bag of Words

- One-Hot Encoding
- Assumption: word order does not matter
- Limitations
  - Discarding word context
  - Discarding grammatical structure
  - Vocabulary inconsistencies (e.g., grammatical errors, conjunctions)
  - Computationally inefficient (sparse matrix with most elements being 0)
- …



gesis Leibniz-Institut für Sozialwissenschaften

Classification Method

Feature Extraction

Document-term matrix

```
In [114]: df2
```

Out[114]:

	aa	aabb	aahl	aaptiv	aaron	aavitsland	ab	ababa	abaca	abad	...
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
564	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
565	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
566	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
567	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...
568	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...

569 rows × 13794 columns

gesis Leibniz-Institut für Sozialwissenschaften

Feature Extraction: Word-embeddings

- Word embeddings capture similarities in words' meaning and function
  - fixed-length & low-dimensional
  - real-valued ("dense") ⇒ word vectors have no zero entries
  - distributed: information about words semantic properties and syntactic functions distributed across dimensions
- Static vs contextual word embeddings

	Static	Contextualized
Representation	static	dynamic
Context-	agnostic	aware
Models	pre-trained, non-adaptable	finetuning
Examples	Word2Vec, GloVe	BERT, GPT-[X]