Quantitative Textanalyse

Sitzung 9: Datenanalyse – Supervised machine learning

Mirko Wegemann

Universität Münster
Institut für Politikwissenschaft

11.12.2024

Plan für heute

- Besprechung Eures Feedbacks
- Einführung in 'statistical learning'
- Anwendung eines prediction tasks in R

Feedback I

Generell eher positives Feedback, **aber**...

- viel Input (hier allerdings auch unterschiedliche Anforderungen)
- zu wenig Zeit für Anwendung
- besseres Zeitmanagement
- teils Unklarheit über Begriffe & Code

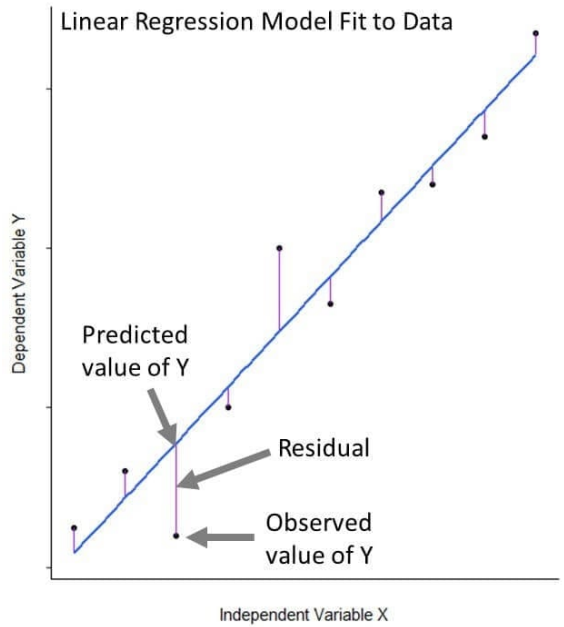
Feedback II

Mein Vorschlag:

- **Nächste Woche**: bringt eure Anwendungsbeispiele sowie Verständnisfragen mit und wir machen eine Lab-Session
- **Nächstes Jahr**: noch zwei Sitzungen zu inhaltlichen Themen, darin: **Embeddings**
- **kein Input** zu Transformer-Modellen

Was meint ihr?

Was ist Machine Learning? I



Grundsätzlich versuchen wir herauszufinden, wodurch wir eine **abhängige Variable** (auch *output* oder *response variable*) bestmöglich mithilfe von **unabhängigen Variablen** (auch *input* oder *predictor variables*) vorhersagen können.

Was ist Machine Learning? II

"In essence, statistical learning refers to a set of approaches for estimating f. In this chapter we outline some of the key theoretical concepts that arise in estimating f, as well as tools for evaluating the estimates obtained."

Zwei Ziele der statistischen Analyse I

1. **Vorhersage** (prediction)
   - Wir wollen ein Ereignis möglichst gut vorhersagen; wir interessieren uns weniger für Erklärungszusammenhänge
2. **Inferenz** (interpretation)
   - Es geht uns weniger darum, ein Ereignis vorherzusagen, als zu analysieren, welche Faktoren es beeinflussen

$\rightarrow$ Fokus im maschinellen Lernen liegt auf der Vorhersage.

Zwei Ziele der statistischen Analyse II

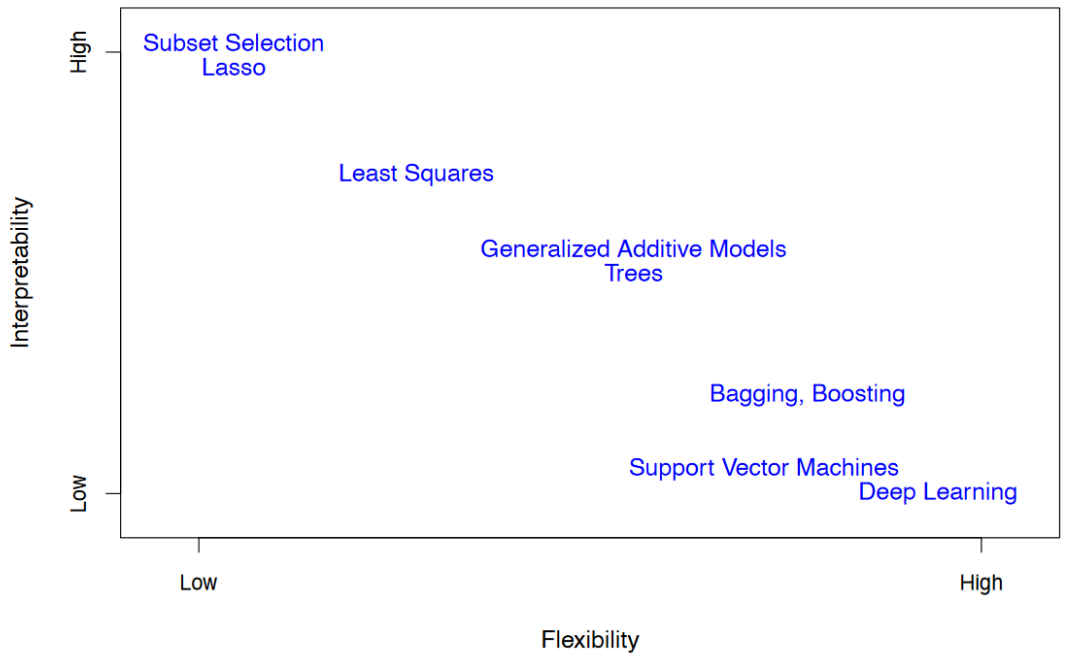


Figure: Zusammenhang zwischen Komplexität (Vorhersagekraft und Interpretabilität (James et al. 2021, p. 25))

Zwei Ziele der statistischen Analyse III

Das hat auch Konsequenzen für die Wahl der optimalen Funktion zur Erklärung unserer abhängigen Variablen. Bei Vorhersage-Problemen interessiert uns weniger die Interpretationsmöglichkeit der Zusammenhänge.

Training- und Test Datensatz I

Für einige Beobachtungen in der Grundgesamtheit kennen wir die Ausprägungen sowohl auf unabhängigen als auch auf der abhängigen Variablen.

- Wir können diese Zusammenhänge nutzen, um die Funktion, welche den Zusammenhang von X auf Y beschreibt, mit möglist geringem Fehler abzubilden
- Ein Teil dieser Daten bildet unseren Trainingsdatensatz, der andere Teil, welcher nicht dazu genutzt wird, unseren Algorithmus zu trainieren, fungiert als Test-Datensatz

Parameter und Komplexität

Wir versuchen, unsere abhängige Variable mithilfe von unabhängigen Variablen zu schätzen.

- Hier kommt wieder Grimmer and Stewart (2013) ins Spiel: "All language models are wrong".
- Es gibt keine Funktion, welche alle Fälle in der Grundgesamtheit korrekt abbildet.
- **Zudem**: Je mehr Parameter wir zum Schätzen von $\hat{Y}$ nutzen, desto eher laufen wir Gefahr, dass wir unsere Daten aus der Stichprobe zwar korrekt vorhersagen, aber Abweichungen in der Grundgesamtheit nicht mehr erklären können (schwache out-of-sample prediction).

Overfitting I

Wenn Matratzenhersterller*innen Overfitting betreiben würden...

Overfitting II

Statistisch lässt sich folgendes beobachten:

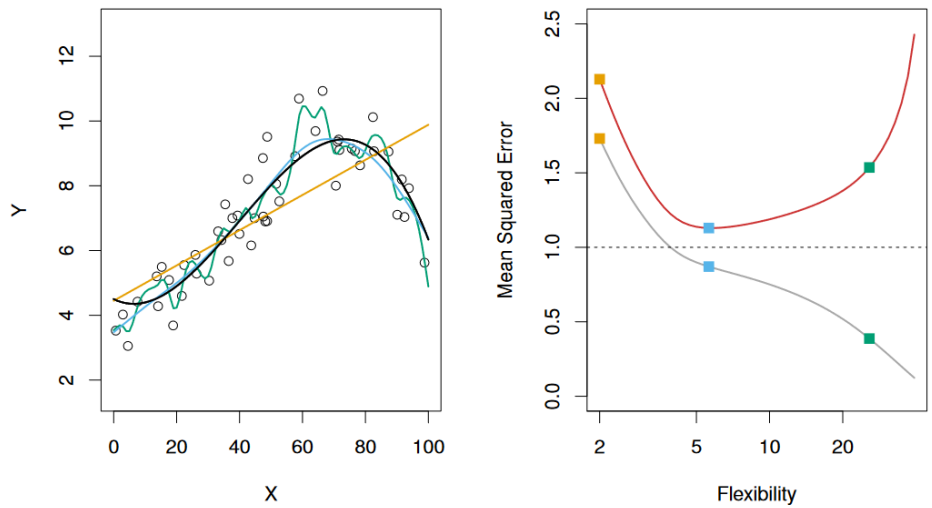


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Figure: Zusammenhang zwischen Flexibilität unseres smoothed lines und dem *mean squared error* im Trainings- und Testdatensatz

Overfitting III

Im maschinellen Lernen begegnen wir stets dem **Variance-Bias-Trade-Off**

- Je komplexer unsere Modelle, desto besser können wir die Werte unserer Beobachtungen auf der abhängigen Variablen vorhersagen → **Komplexität reduziert Bias**
- Je komplexer unsere Modelle, desto mehr hängen sie von den Daten ab, mit denen wir sie trainiert haben → **Komplexität erhöht Varianz**

Unser Ziel? Den Sweetspot finden, in dem Bias und Varianz minimiert werden.

Regression vs. Classification I

Wir können im *machine learning* zwischen **metrischen** und **kategorialen** Variablen unterscheiden

- Bei **metrischen** Variablen wollen wir einen konkreten Wert vorhersagen
  - Bsp.: Wie viel Einkommen kann eine Person erwarten, welche einem Beruf in einem bestimmten Land ausübt?
  - Schätzung oftmals mit **ordinary least squares**

Regression vs. Classification II

- Bei **kategorialen** Variablen geht es uns um die Bestimmung einer Kategorie
  - Bsp.: Handelt es sich bei einem Text mit den Features w um einen Text zu Kategorie K?
  - Schätzung oftmals durch logistische Regression
  - auch hier erhalten wir in der Regel einen metrischen Wert (i.d.R. die Wahrscheinlichkeit, dass ein Fall zu Kategorie "K" gehört); mithilfe von Entscheidungsregel bestimmen wir die Kategorie

$\rightarrow$ für Textanalysen sind wir meist an Klassifikation interessiert

Predicting text

Manchmal kennen wir die Kategorien eines Teils unseres Korpus bereits, möchten aber die Kategorien für die übrigen Texte vorhersagen.
Lösung: **supervised models**!

- In unserem Anwendungsbeispiel werden wir MARPOR-Daten verwenden, welche bereits durch Expert*innen klassifiziert worden sind
- Eine andere gängige Anwendung sind Sentiment-Analysen

Ziel: Auf Basis annotierter Daten Beziehungen in nicht gelabelten Daten ableiten.

Grundlegende Konzepte I

- **Goldstandard**: Um unsere Modelle zu trainieren, benötigen wir annotierte Daten; diese gelten als unser Goldstandard; die Performance unserer Modelle vergleichen wir mit diesen Daten, sie können aber nie besser sein.
- **Trainings-/Testdaten**: Wir teilen unsere Daten immer mind. in Trainings-/Testdaten (besser 3er-Split mit Validierungsdaten)
  1. Trainingsdaten dienen dazu, unseren Algorithmus an den prediction task anzupassen; auf Basis dieser Daten werden Input-Vektoren (Wörtern) Gewichte zugeordnet
  2. Testdaten werden genutzt, um zu evaluieren, ob unser Modell auch Dateien außerhalb des Trainingssamples vorhersagen kann

Grundlegende Konzepte II

- **Ziel**: Unser Ziel ist es, nicht-gelabelte Texte richtig vorherzusagen.
- **Klassifikationsalgorithmus**: Modell zur Klassifikation unserer Daten (binär, multi-class, multi-label).
- **Performance-Metriken**: Accuracy, F1-Score, True-/False-Positive-Rates zur Bewertung der Model-Performance.

Klassifikationsalgorithmen I

Naive Bayes

- Familie von Klassifikationsalgorithmen, die auf dem Bayes-Theorem basieren.
- Wahrscheinlichkeit wird durch das Auftreten eines Ereignisses beurteilt.
- Beispiel: Wie wahrscheinlich ist es, dass jemand über Migration spricht, wenn die Begriffe ‘Flüchtlinge’, ‘sind’, ‘willkommen’, ‘hier’, ‘.’ verwendet werden?
- Geht von Merkmalsunabhängigkeit aus (behandelt jedes Merkmal unabhängig) $\rightarrow$ im obigen Beispiel hat der Begriff ‘Flüchtlinge’ nichts mit ‘willkommen’ zu tun.
- Jedes Merkmal wird gleich gewichtet $\rightarrow$ ‘hier’ ist genauso wichtig wie ‘Flüchtlinge’.

Klassifikationsalgorithmen II

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes

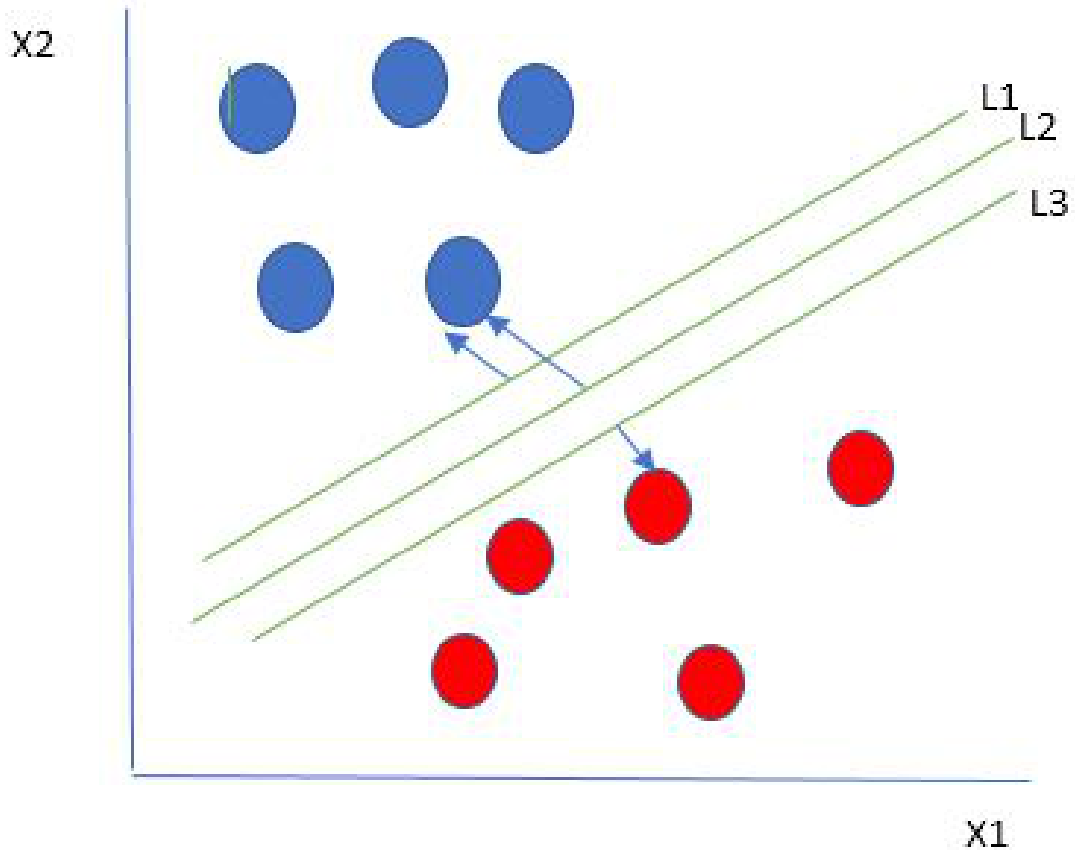
Grafik und Intuition

Klassifikationsalgorithmen III

Support Vector Machines

- Grundidee: Eine *hyper plane* (decision boundary) zeichnen, die unsere Daten optimal trennt.
- Support-Vektoren sind die Fälle, die der hyper plane am nächsten sind.
- Ziel ist es, den Abstand zwischen den empirischen Fällen der Klassen zu maximieren.

Klassifikationsalgorithmen IV



Grafik und Intuition

Daten

Unsere annotierten Daten sollten mindestens zwei Spalten umfassen:

1. **feature**: Ein oder mehrere Input Vektoren (z. B. Wörter in einem Text).
2. **label**: Eine Variable, die die Kategorie des Merkmals angibt.

In der Notation von Regressionsanalysen: Das Label ist Y, und das Merkmal ist X; wir versuchen Y durch X_i zu erklären.

Klassifikation in R

- Heute: `quanteda.textmodels` → Naive Bayes und SVM
- Im nächsten Jahr: supervised machine learning mithilfe von embeddings in deep neural networks

Vorbereitung

Wenn wir eine `dfm` mit Labels als `docvars` haben, müssen wir die Matrix in Trainings- und Testdaten aufteilen (Ratio ist eure Entscheidung, aber oft wird 80/20 gewählt).
Da wir Trainings- und Testsplits kreieren, müssen wir diejenigen Merkmale (Wörter) entfernen, die nicht in beiden unserer `dfm` vorhanden sind.

```
> train <- dfm_sample(m_dfm_sub,0.8*nrow(df_sub))
> test <- dfm_subset(m_dfm_sub,
+   !(docnames(m_dfm_sub) %in% docnames(train)))
> train <- train %>% dfm_trim(1)
> test <- dfm_match(test, featnames(train))
```

Modellschätzung

Im nächsten Schritt können wir bereits unseren Algorithmus trainieren.

```
1  > nb_model<-textmodel_nb(train,docvars(train,
       "label"))
2  > nb_model
3
4  Call:
5  textmodel_nb.dfm(x = train, y = docvars(train,
       "label"))
6
7  Distribution: multinomial ; priors: 0.5 0.5 ;
       smoothing value: 1 ; 18796 training documents;
       fitted features.
```

Modellinferenz

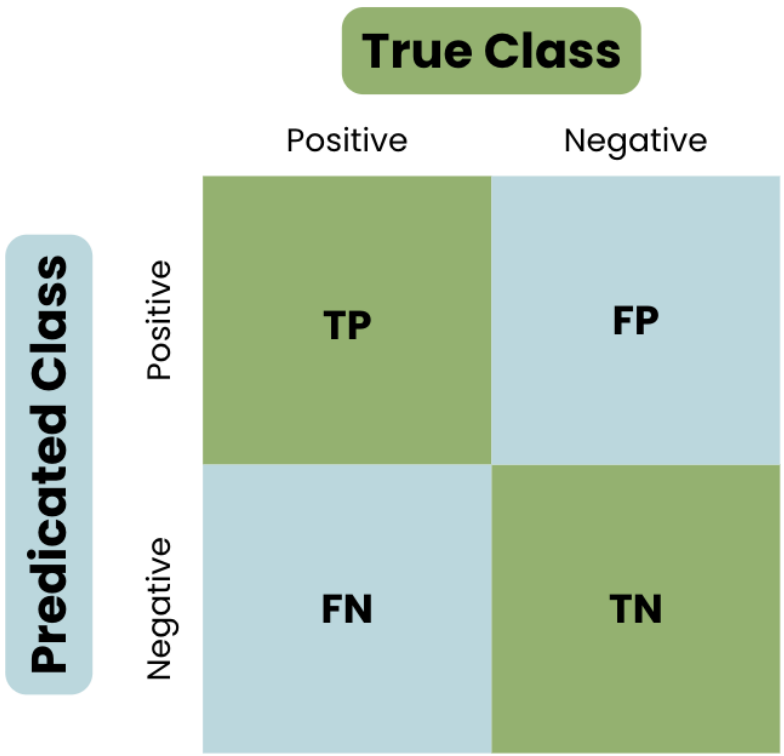
Nachdem wir das Modell trainiert haben, können wir es für Inferenz verwenden (hier für unsere Testdaten, um die Leistung zu bewerten, aber potenziell auch für out-of-sample-Daten).

```
> test_predictions<-predict(nb_model, newdata=test)
```

Let's do it in R!

Evaluation I

Da wir bereits annotierte Daten haben, ist die Evaluation einfacher als bei unsupervised approaches. Es gibt verschiedene Metriken, die wir berücksichtigen können:



Evaluation II

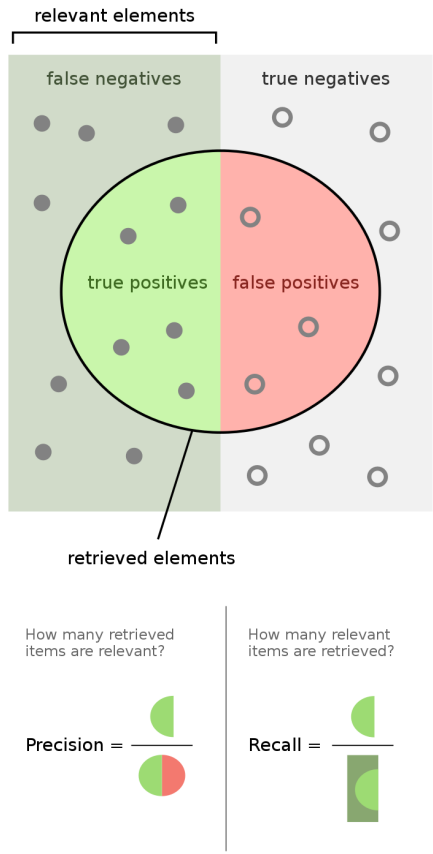
Accuracy

$$\frac{TP + TN}{N} \tag{1}$$

Wie viele Fälle haben wir korrekt vorhergesagt?

Evaluation III

Precision und Recall (Sensitivität)



Evaluation IV

Die **Spezifizität** ist das Gegenteil der Sensitivität (TN/(TN+FP))

Evaluation V

F1-Score:
Unsere Performance Metrics hängen von der Balance unseres Datensatzes ab (wie viele Fälle haben wir in der jeweiligen Klasse). Um dies miteinzubeziehen verwenden wir häufig den F1-Score:

$$2 \times \frac{Precision * Recall}{Precision + Recall} \tag{2}$$

Evaluation V

Danach können wir eine Matrix erstellen, die aus empirischen und vorhergesagten Werten besteht, um unsere Leistung zu überprüfen (Konfusionsmatrix).

```
> (eval_mat <-
    table(docvars(test,"label"),test_predictions))
test_predictions
0    1
0  396   50
1  239 4014
```

Evaluation VI

Das caret-Package bietet weitere Klassifikationsmetriken.

```
> confusionMatrix(test_predictions,
    as.factor(docvars(test,"label")))
[...]
Accuracy : 0.9385
[...]
Sensitivity : 0.88789
Specificity : 0.94380
Balanced Accuracy : 0.91585
```

Bot Detection on Russian Political Twitter I

- **Forschungsfrage**:
- **Methodologie**:
- **Ergebnisse**:
- **Implikationen**:

Bot Detection on Russian Political Twitter II

- **Forschungsfrage**: Wie können wir am besten Bots auf Twitter identifizieren?
- **Methodologie**: Ensemble-Classifier mit hoher Precision und akzeptablem Recall
- **Ergebnisse**: Zur Zeit des Erscheinen des Artikels sind oftmals mehr als 50% der Tweets von Bots geschrieben; meist zu russischen Nachrichten
- **Implikationen**: Vorsicht im Umgang mit Social Media

Bot Detection on Russian Political Twitter III

Zur Methode:

- Download von Twitter-Daten über die Twitter API (RIP)
- 3,130 russische Accounts wurden von 50 Coder*innen annotiert, u.a. deren Interaktionen mit anderen Nutzer*innen, bibliographische Infos, Fotos
- **nur 1,000 Accounts** werden als Trainingsdaten ausgewählt, in denen es eine hohe Intercoder-Reliabilität gibt [Problem?]
- vier verschiedene Klassifikationsalgorithmen; five-fold cross validation
- Klassifikationsregel: Einstimmigkeit der Algorithmen
- Anwenden des Modells auf nicht-annotierte Accounts

Bot Detection on Russian Political Twitter IV

Table 2. Performance metrics over 10 test sets

	Precision	*Recall*	*Specificity*
Ridge logistic regression	0.92	0.87	0.92
SVM (RBF kernel)	0.90	0.84	0.90
XGBoost	0.87	0.91	0.87
SAMME	0.92	0.89	0.92
Ensemble (unanimous voting: bots)	0.99	0.77	0.99
Ensemble (unanimous voting: non-bots)	0.93	0.79	0.94

Entries are performance metrics averaged over 10 test sets. $Precision = Pr(\widehat{bot}|bot)$. $Recall = Pr(\widehat{bot}|bot)$. $Specificity = Pr(\widehat{nonbot}|nonbot)$.

Source: Authors' calculations based on data collected from Twitter API.

RBF, radial basis function; SAMME, Stagewise Additive Modeling using Multiclass exponential loss function; SVM, support vector machine.

Figure: Ergebnisse der Klassifikationsalgorithmen

Was bedeuten die Ergebnisse?

Bot Detection on Russian Political Twitter V

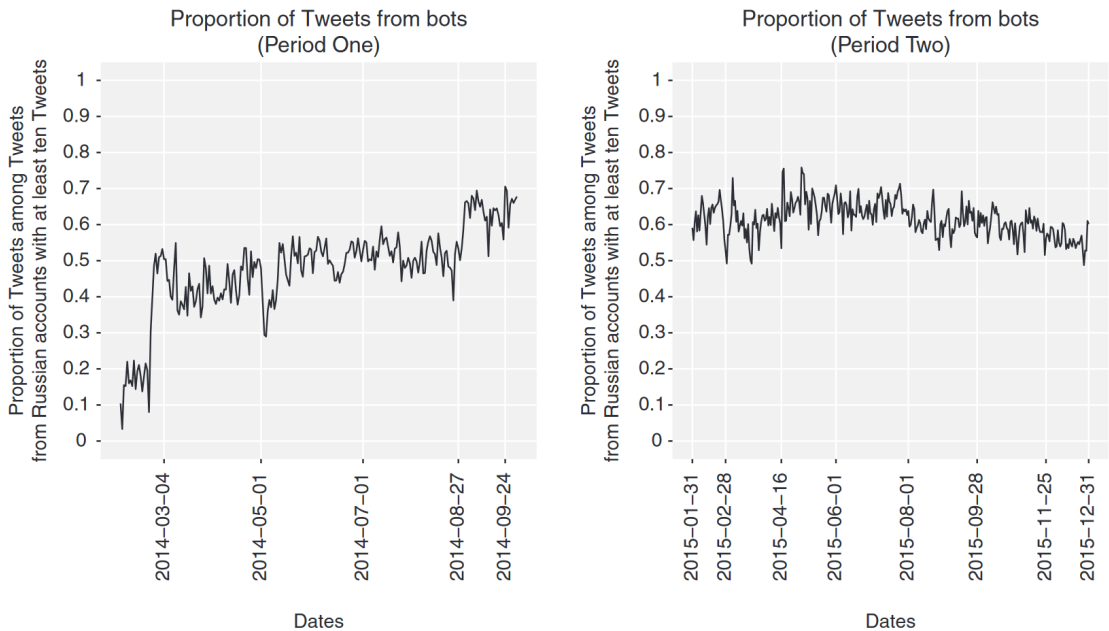


Figure: Tweets von russischen Bots im russischen Twitter

Bot Detection on Russian Political Twitter VI

Table 3. Top 20 features (ridge logistic regression)

Feature	*Coefficient Sign*[a]	*Feature Importance*[b]
Platform: Twitter for Websites	+	1.3
Platform: Tweet Button	+	5.0
Platform: Twitter for iPhone	−	5.9
Platform: dlvr.it	+	7.2
Platform: Twitter for Android Tablets	+	7.2
% of Retweets	+	8.9
% of Tweets with a hashtag	−	9.0
Platform: Twitter for Android	−	9.9
Platform: Twitter for iPad	−	10.3
Platform: Mobile Web (M2)	−	10.8
Geo-enabled Tweets (binary)	−	11.0
Platform: Mobile Web (M5)	−	11.3
Platform: Twitter website	−	13.4
Politicalness[c]	−	14.0
Platform: Twitterfeed	+	14.7
Change of default profile image (binary)	+	15.0
% of Tweets at somebody	−	16.2
Platform: ifttt.com	−	17.8
Platform: Twitter Web Client	−	18.6
Entropy of platform use	−	18.7

Figure: Die wichtigsten Input-Variablen für die Detection der Bots

Ausblick

- **nächste Woche:** Lab → bringt eure Ideen für Anwendungsbeispiele mit

Literatur

Grimmer, J., & Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, *21*(3), 267–297. https://doi.org/10.1093/pan/mps028

James, G., Witten, D., Hastie, T., & Tibshirani. (2021). *An Introduction to Statistical Learning*. Retrieved September 19, 2024, from https://www.statlearning.com