Text as Data

Session 9: Supervised machine learning

Mirko Wegemann

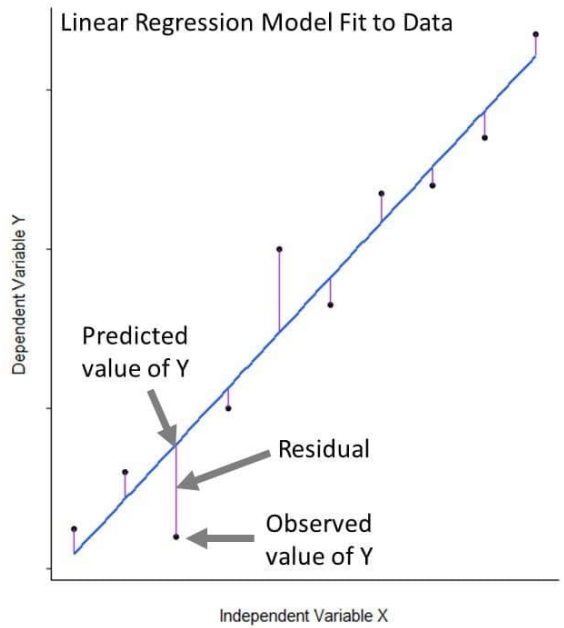
Universität Münster
Institut für Politikwissenschaft

01 July 2026

Plan for today

- Presentation by Jonas
- Introduction to 'statistical learning'
- Application of a prediction task in R

What is Machine Learning? I



Basically, we are trying to find out how we can best predict a **dependent variable** (also *output* or *response variable*) using **independent variables** (also *input* or *predictor variables*).

Two goals of statistical analysis I

1. **Prediction** (prediction)
   - We want to predict an event as accurately as possible; we are less interested in explanatory relationships.
2. **Inference** (interpretation)
   - Our aim is less to predict an event than to analyze which factors influence it.

→ The focus in machine learning is on prediction.

Two goals of statistical analysis II

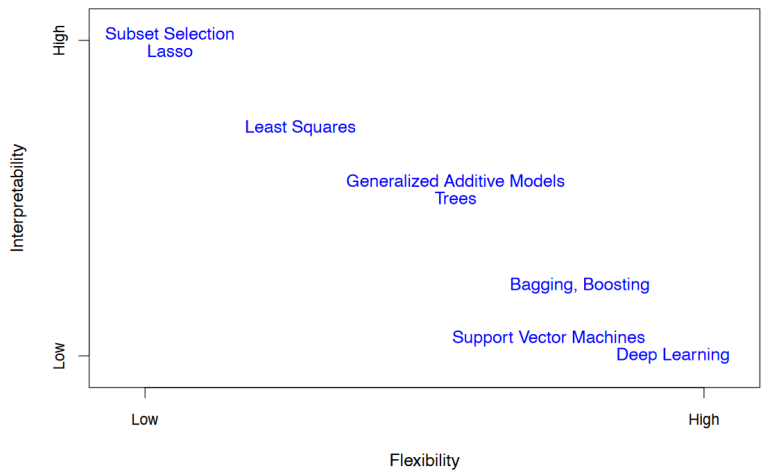


Figure: Relationship between complexity (predictability and interpretability (James et al. 2021, p. 25))

There is a trade-off between prediction and inference.

Training and test dataset I

For some observations in the population, we know the values on both the independent and the dependent variables.

- We can use these relationships to map the function that describes the relationship between X and Y with the smallest possible error.
- Part of this data forms our training dataset, the other part, which is not used to train our algorithm, serves as a test dataset.

Parameters and Complexity

We try to estimate our dependent variable using independent variables.

- Here (Grimmer and Stewart 2013) comes into play again: "All language models are wrong".
- There is no function that accurately represents all cases in the population.
- Furthermore: The more parameters we have to estimate $\hat{Y}$, the greater the risk that we may correctly predict our data from the sample, but not the data outside our sample

Overfitting I

If mattress manufacturers practiced overfitting...

Overfitting II

Statistically, the following can be observed:

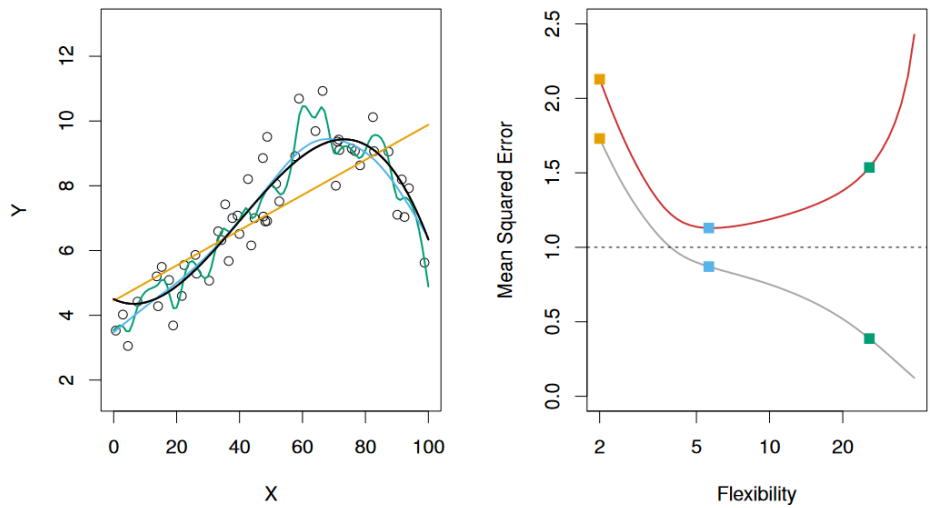


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed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

Figure: Relationship between the flexibility of our smoothed lines and the *mean squared error* in the training and test dataset

Overfitting III

In machine learning, we constantly encounter the **Variance-Bias-Trade-Off**.

- The more complex our models, the better we can predict the values of our observations on the dependent variable. → Complexity reduces bias
- The more complex our models, the more they depend on the data we used to train them. → Complexity increases variance

Our goal? To find the sweet spot where bias and variance are minimized.

Regression vs. Classification I

In machine learning, we can distinguish between metric and categorical variables.

- For metric variables, we want to predict a specific value.
  - Example: How much income can a person expect who works in a particular country?
  - Estimation often using **ordinary least squares** (OLS)

Regression vs. Classification II

- With categorical variables, we are concerned with determining a category.
  - Example: Can a text with the features w be classified into category K?
  - Estimation is often done using logistic regression.
  - Here too, we usually obtain a metric value (generally the probability that a case belongs to category "K"); using decision rules, we determine the category.

$\rightarrow$ For text analysis, we are usually interested in classification.

Predicting text

Sometimes we already know the categories for part of our corpus, but want to predict the categories for the remaining texts.
Solution: **supervised models**!

- In our application example, we will use MARPOR data that has already been classified by experts.
- Another common application is sentiment analysis.

Goal: To derive relationships in unlabeled data based on annotated data.

Basic Concepts I

- **Gold standard**: To train our models, we need annotated data; this is considered our gold standard; we compare the performance of our models to these data, but they can never be better.
- **Training/Test Data**: We always divide our data into at least training/test data (better a 3-way split with validation data)
  1. Training data is used to adapt our algorithm to the prediction task; based on this data, weights are assigned to input vectors (words).
  2. Test data is used to evaluate whether our model can also predict files outside the training sample.

Basic Concepts II

- **Target**: Our goal is to correctly predict unlabeled texts.
- **Classification algorithm**: Model for classifying our data (binary, multi-class, multi-label).
- **Performance metrics**: Accuracy, F1 score, True/False Positive Rates for evaluating model performance.

Classification algorithms I

Naive Bayes

- Family of classification algorithms based on Bayes' theorem.
- Generative model: Probability is assessed by the occurrence of features (based on highest probability) $\rightarrow$ it tries to understand how classes look like and then applies probability to each feature Example: How likely is it that someone will talk about migration if the terms 'Refugees', 'are', 'welcome', 'here', '.' be used?
- Assumes feature independence (treats each feature independently) $\rightarrow$ In the example above, the term 'welcome' has always the same predictive value, irrespectively of it being used next to 'refugees' or 'home'.
- Model multiplies probabilities of each feature rather than interacting them

Classification algorithms 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

Graphics and intuition

Classification algorithms III

Logistic Regression

- Basic idea: Predict an outcome through several features
- The advantage over naive bayes: no independence assumption, so that feature correlations can be taken into account
- Logistic regression is discriminative and tries to optimize weights to distinguish classes

Classification algorithms IV

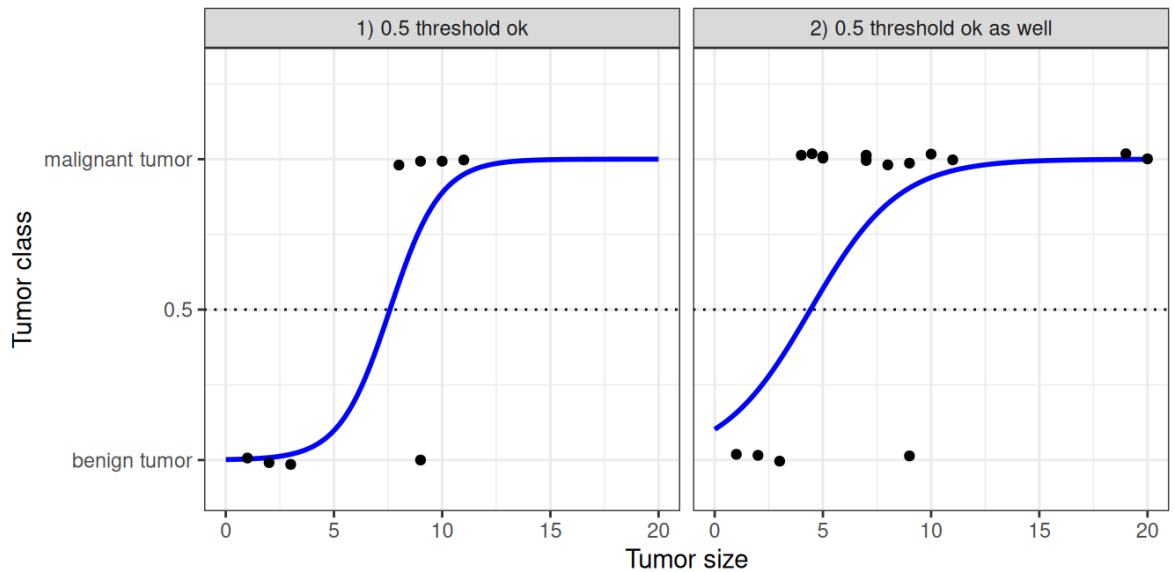


Figure: Decision rule in logistic regression, Source

Data

Our annotated data should include at least two columns:

1. **feature**: One or more input vectors (e.g., words in a text).
2. **label**: A variable that specifies the category of the feature.

In the notation of regression analyses: The label is Y, and the characteristic is X; we try Y through X_i to explain.

Classification in R

- Today: `quanteda.textmodels` → Naive Bayes and SVM
- If you want to dive into supervised machine learning: use embeddings in deep neural networks or transformer models

Preparation

If we have a `dfm` with labels as `docvars`, we need to split the matrix into training and test data (the ratio is up to you, but 80/20 is often chosen).
Since we are creating training and test splits, we need to remove those features (words) that are not present in both of our `dfm`.

```
> 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))
```

Model estimation

In the next step, we can already train our algorithm.

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

Model Inference

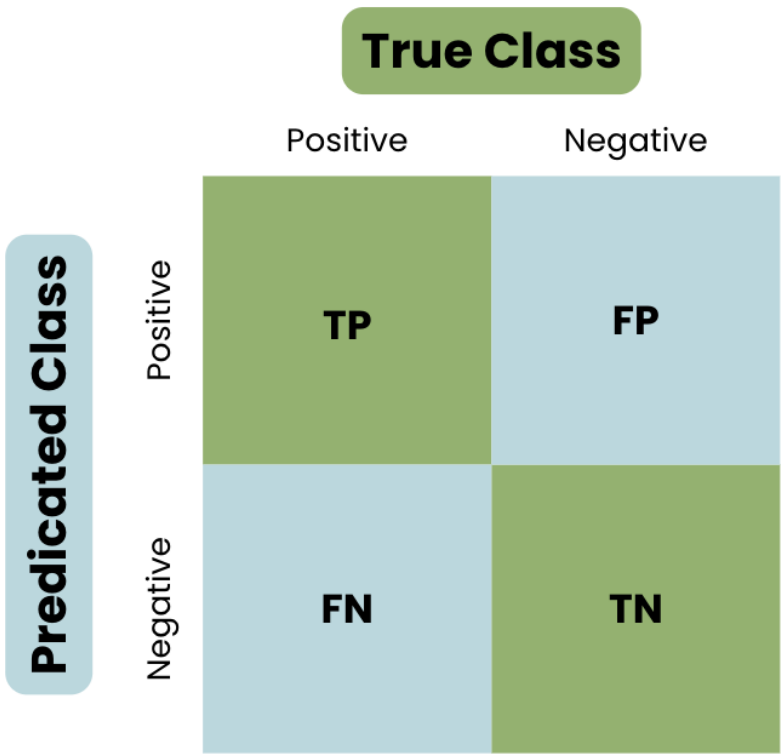
Once we have trained the model, we can use it for inference (here for our test data to evaluate performance, but potentially also for out-of-sample data).

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

Let's do it in R!

Evaluation I

Since we already have annotated data, the evaluation is simpler than with unsupervised approaches. There are several metrics we can consider:



Evaluation II

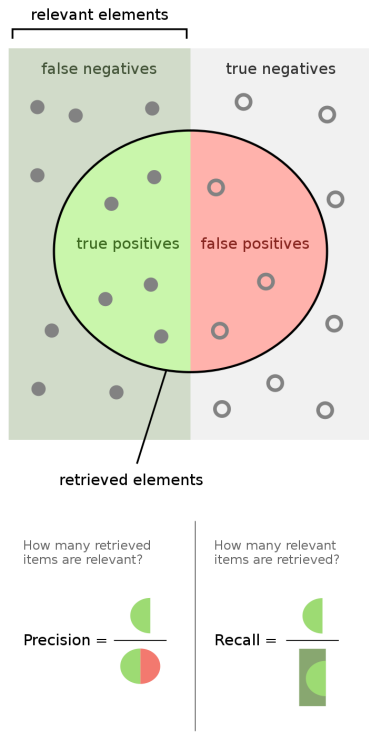
Accuracy

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

How many cases did we predict correctly?

Evaluation III

Precision and Recall (Sensitivity)



Specificity is the opposite of sensitivity (TN/(TN+FP)).

Evaluation IV

F1-Score:
Our performance metrics depend on the balance of our dataset (how many cases we have in each class). To incorporate this, we often use the F1 score:

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

Evaluation V

After that, we can create a matrix consisting of empirical and predicted values to check our performance (confusion matrix).

```
1 > (eval_mat <-
      table(docvars(test,"label"),test_predictions))
2 test_predictions
3 0 1
4 0 396 50
5 1,239,4014
```

Evaluation VI

The caret package offers additional classification metrics.

```
> 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

- **Research question**:
- **Methodology**:
- **Results**:
- **Implications**:

Bot Detection on Russian Political Twitter II

- **Research question**: What is the best way to identify bots on Twitter?
- **Methodology**: Ensemble classifier with high precision and acceptable recall
- **Results**: At the time of publication, often more than 50% of tweets are written by bots; mostly about Russian news.
- **Implications**: Caution when using social media

Bot Detection on Russian Political Twitter III

On method:

- Download Twitter data via the Twitter API
- 3,130 Russian accounts were annotated by 50 coders, including their interactions with other users, bibliographic information, and photos.
- Only 1,000 accounts are selected as training data where there is high intercoder reliability [Problem?]
- four different classification algorithms; five-fold cross validation
- Classification rule: Unanimity of algorithms
- Applying the model to non-annotated 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: Results of the classification algorithms

What do the results mean?

Bot Detection on Russian Political Twitter V

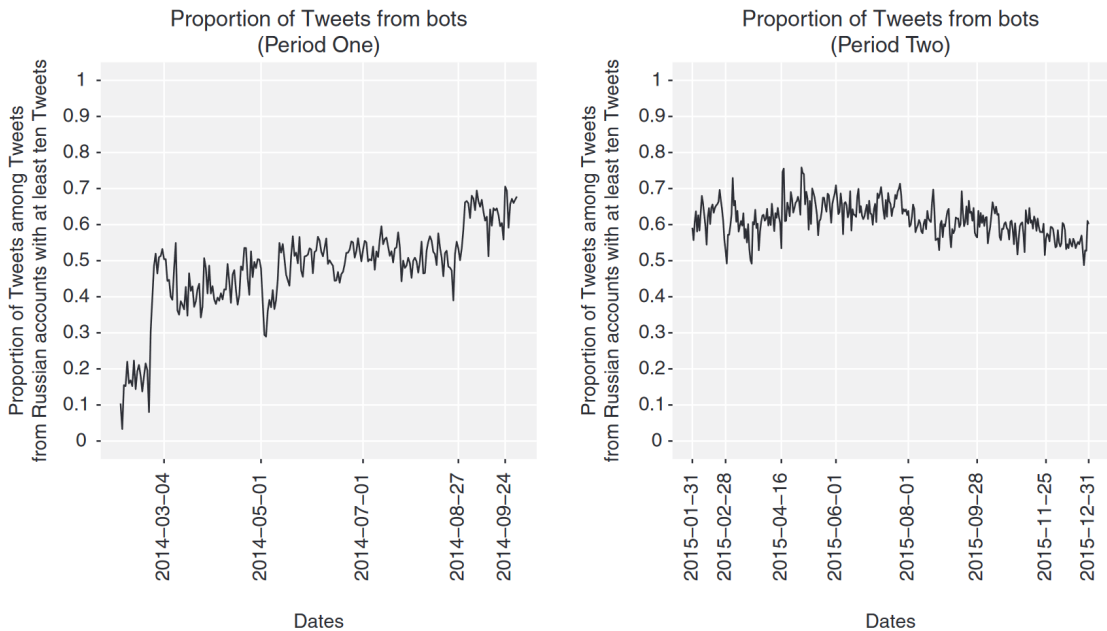


Figure: Tweets from Russian bots on Russian Twitter

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: The most important input variables for bot detection

Outlook

- **Next week:** Embeddings
- We will learn a new representation of text that captures context better
- Literature
  - Rodriguez, P., & Spirling, A. (2021).Word Embeddings: What works, what doesn't, and how to tell the difference for applied research. *The Journal of Politics*. https://doi.org/10.1086/715162
  - Rodriguez, P. L., Spirling, A., & Stewart, B. M. (2023).Embedding Regression: Models for Context-Specific Description and Inference. *American Political Science Review*, *117*(4), 1255–1274. https://doi.org/10.1017/S000305542200122 8 (complementary)

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

Rodriguez, P., & Spirling, A. (2021).Word Embeddings: What works, what doesn't, and how to tell the difference for applied research. *The Journal of Politics*. https://doi.org/10.1086/715162

Rodriguez, P. L., Spirling, A., & Stewart, B. M. (2023).Embedding Regression: Models for Context-Specific Description and Inference. *American Political Science Review*, *117*(4), 1255–1274. https://doi.org/10.1017/S0003055422001228