Masterarbeit

Vortrag 27.11.2008

TECHNISCHE UNIVERSITÄT DARMSTADT

Thunderbird-Plugin zur Erkennung anhangverdächtiger E-Mails

Marco Ghiglieri

Prof. Dr. Johannes Fürnkranz

Agenda

TECHNISCHE UNIVERSITÄT DARMSTADT

- Motivation
- Algorithmen
  - Einfacher Algorithmus
  - Erweiterter Algorithmus
  - Naive Bayes
  - Paul Graham
- Evaluierung
- Fazit/Ausblick

Motivation

TECHNISCHE UNIVERSITÄT DARMSTADT

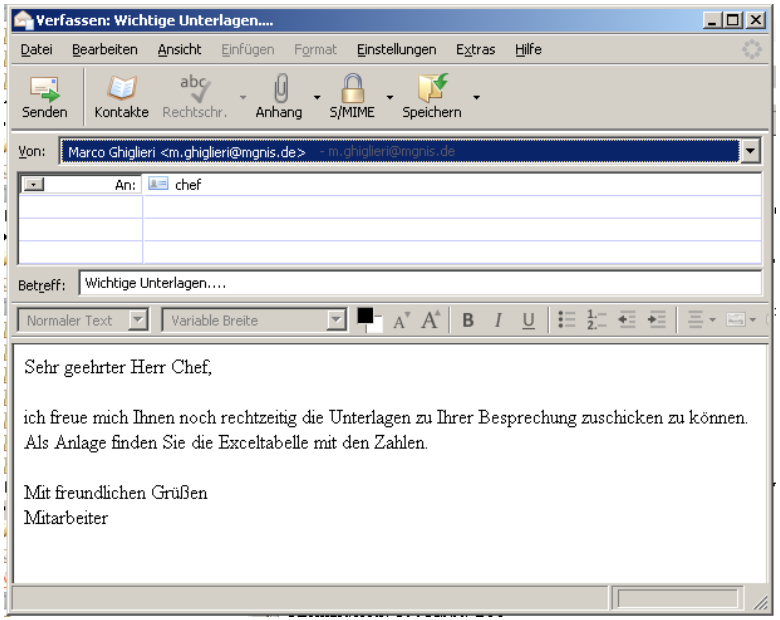
- 210.000.000.000 E-Mails pro Tag*
- 70.000.000.000 „saubere“ E-Mails pro Tag
- 700.000.000 Attachments pro Tag
- 7.000.000 davon vergessen

*Quelle:http://email.about.com/od/emailtrivia/f/emails_per_day.htm

Situation 80mal/Sekunde

Gemalt von Jane Elsemüller

15033	57342	48	390
15426	57620	50	393
15533	57691	13	107
16330	58317	99	797
16541	58445	20	211
16889	58670	32	348
17485	59043	64	596
17882	59302	43	397
19615	60421	189	1733



Situation 80mal/Sekunde

TECHNISCHE UNIVERSITÄT DARMSTADT

Gemalt von Jane Elsemüller

Problemstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mozilla Thunderbird-Plugin
- Erkennen von E-Mails, die womöglich ein Attachment haben sollten.
- Viele richtige Klassifikationen

mozilla Thunderbird™

Spam- und Attachment Erkennung

TECHNISCHE UNIVERSITÄT DARMSTADT

Spam

- Unerwünschte E-Mails
- Klassifikation einfacher
- Zweiklassenproblem
- Benutzerspezifische & allgemeine Merkmale
- Header und Text
- Böswillig

Attachments

- Erwünschte E-Mails
- Klassifikation schwierig
- Zweiklassenproblem
- Benutzerspezifische Merkmale
- Nur Text
- Vergesslichkeit

➔ Ähnliches Klassifikationsproblem

Einfacher Algorithmus Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

E-Mail 1:

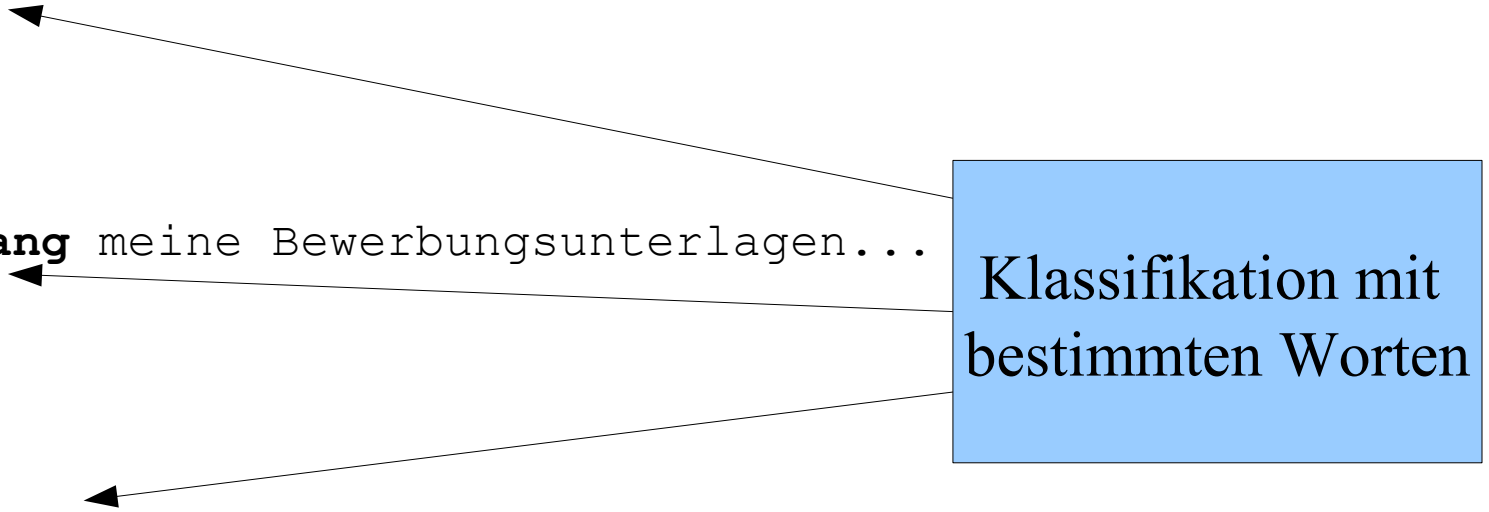
`... im **Anhang** sind die Urlaubsbilder...`

E-Mail 2:

`...im **Anhang** meine Bewerbungsunterlagen...`

E-Mail 3:

`...als **Anlage** die neuesten Zahlen...`



➔ **words=[anhang,anlage]**

Einfacher Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

```
words = [anhang, anlage]
for each token in email:
        if (lowercase of token) in words:
                showAlert()
                return
```

- Gmail attachment reminder
- Umgesetzt in einigen Thunderbird-Plugins
  - Attachment Reminder (Mozilla-ID: 5759)
  - Check and Send (Mozilla-ID: 2281)

Zu finden unter: http://userscripts.org/scripts/review/2419 oder https://addons.mozilla.org/de/thunderbird/addon/xxxx (für xxxx die vierstellige Nummer einsetzen)

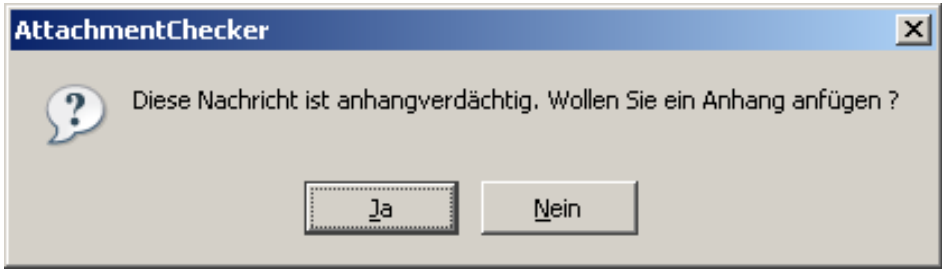
Einfacher Algorithmus Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

E-Mail:

```
Als Anlage finden Sie die Exceltabelle mit den Zahlen.
```

1. Prüfen von „Als“ ==> weiter
2. Prüfen von „Anlage“ ==> Popup




```
words = [anhang,anlage]
for each token in email:
      if (lowercase of token) in words:
            showAlert()
            return
```

Einfacher Algorithmus
Vor- und Nachteile

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorteile

- Einfach
- Nachvollziehbar
- Schnell
- Platzsparend

Nachteile

- Sprachabhängig
- Benutzerpflege
- Lernt nicht
- Keine Wortabhängigkeiten
- Zuviele Fehlalarme

➔ **Erweiterter Algorithmus**

Worthäufigkeiten

TECHNISCHE UNIVERSITÄT DARMSTADT

- Alle weiteren Algorithmen beachten nun die Worthäufigkeiten

Att	
anlage	4
bilder	2
...	

Noatt	
hallo	40
bild	11
...	

Att = Menge aller Tupel (Wort, Anzahl) in E-Mails mit Attachments

Noatt = Menge aller Tupel (Wort, Anzahl) in E-Mails ohne Attachments

- Beeinflussen die Klassifikation

Erweiterter Algorithmus - Idee

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wortliste `words` automatisch füllen
  - Wenn Anzahl des Wortes in att größer ist als Anzahl des Wortes in noatt ==> Füge Wort hinzu
  - Wenn Anzahl des Wortes in noatt größer ist als Anzahl des Wortes in att ==> Lösche Wort
  - Eine Startliste ist möglich
- Die Erkennung verläuft wie beim einfachen Algorithmus

Erweiterter Algorithmus Lernalgorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

```
for each word in email:
      if email has_attachment then
            count word in email add to att_list(word)
      else
            count word in email add to noatt_list(word)

      for each token in att_list:
            if 2*att_list[token]> noatt_list[token]
                  insert token in words

      for each token in words:
            if noatt_list[token]>att_list[token]
                  remove token from words
```

Erweiterter Algorithmus Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gegeben sind Noatt, Att, words

Att

anhang	50
anlage	70
bilder	3
unterlagen	5

Noatt

unterlagen	5
hallo	70
einkaufen	3
party	5

words=[anhang,anlage, unterlagen]

- Neue E-Mail ohne Attachment: `...Unterlagen...`

Noatt

unterlagen	**6**
hallo	70
einkaufen	3
party	5

➔ **words=[anhang,anlage]**

Erweiterter Algorithmus
Vor- und Nachteile

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorteile

- Einfach
- Schnell

Nachteile

- Keine Wortabhängigkeiten
- Zuviele Fehlalarme
- Undurchsichtig für den Benutzer

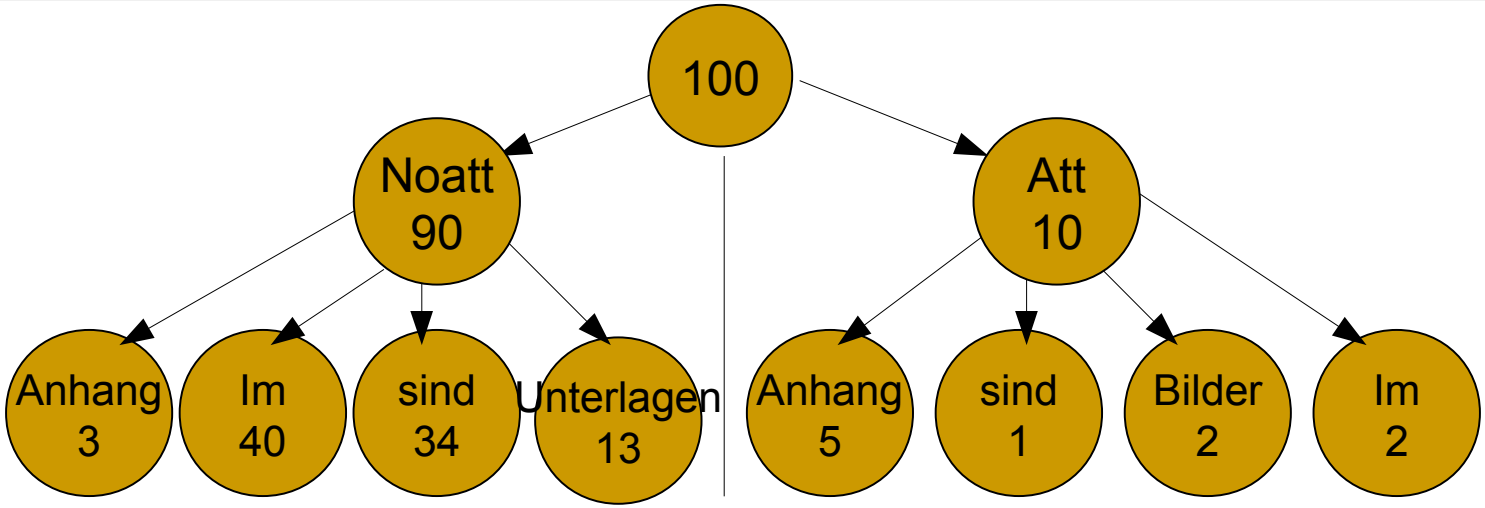
Naive Bayes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Annahme: Auftreten von Wörtern ist in E-Mails unabhängig voneinander
  - Dies ist natürlich nicht der Fall
- Wahrscheinlichkeitsbasiert
- Erkennungsraten beim Spam-Problem hoch

Naive Bayes Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT



Neue E-Mail:

`...Im Anhang sind Bilder...`

$$p(E|c) = p(c) \cdot \prod_{i=0}^{W} p(w_i|c)$$

$$p(E|Noatt) = \frac{90}{100} \cdot \frac{40}{90} \cdot \frac{3}{90} \cdot \frac{34}{90} \cdot \frac{1}{100} = 0.00005 \quad \Rightarrow 20\%$$

$$p(E|Att) = \frac{10}{100} \cdot \frac{2}{10} \cdot \frac{5}{10} \cdot \frac{1}{10} \cdot \frac{2}{10} = 0.0002 \quad \Rightarrow 80\%$$

Naive Bayes
Probleme

TECHNISCHE UNIVERSITÄT DARMSTADT

- Wahrscheinlichkeit für Wort, dass nicht vorkommt
- Sehr kleine Wahrscheinlichkeiten verursachen Unterlauf

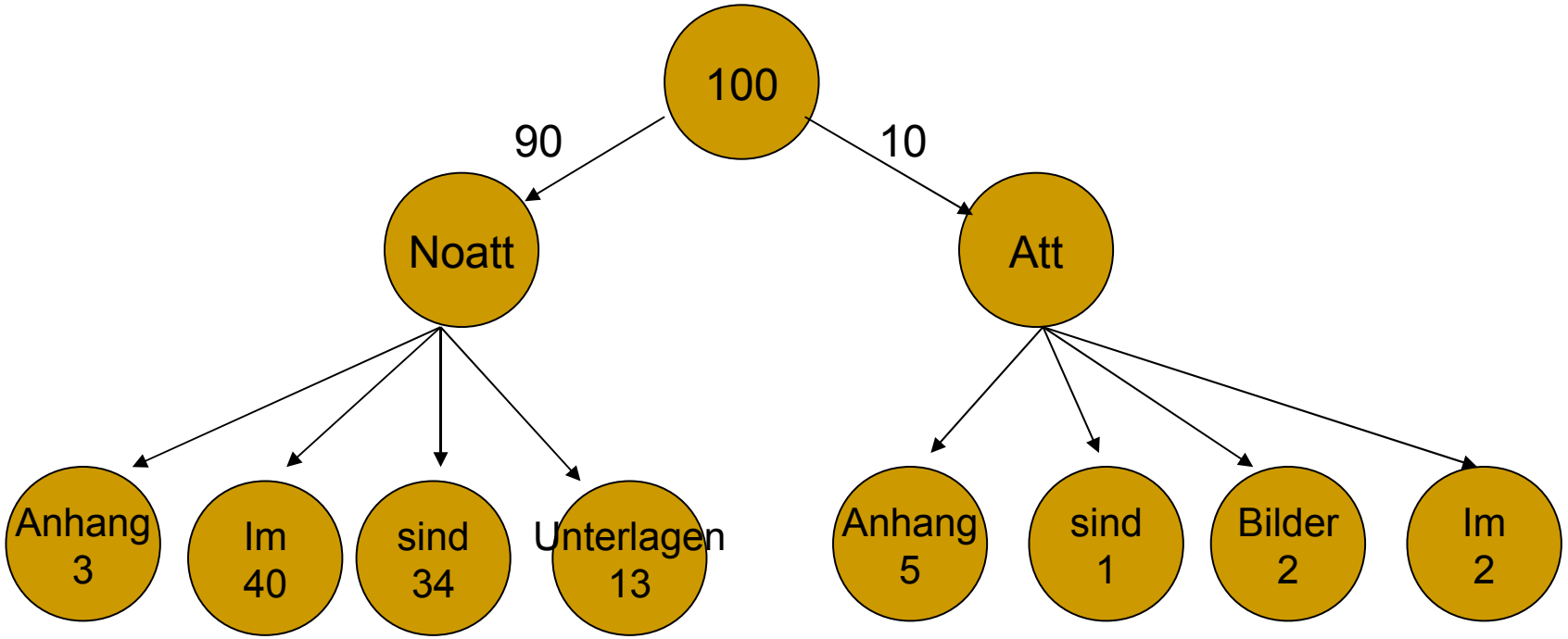
Paul Grahams Algorithmus

TECHNISCHE UNIVERSITÄT DARMSTADT

- Erkennung geschieht mit Pseudowahrscheinlichkeiten
- Sehr gute Erkennungsraten beim Spam-Problem
- Eingesetzt in Thunderbird als Spamfilter

Paul Graham Beispiel

TECHNISCHE UNIVERSITÄT DARMSTADT



10 E-Mails wurden erfasst:

9 ohne Attachment

1 mit Attachment

Paul Graham Beispiel
Berechnung des Gewichts

TECHNISCHE UNIVERSITÄT DARMSTADT

Berechnung des Gewichts des Wortes „anhang“:

$$n_{anhang,att}=5$$
$$n_{anhang,noatt}=3$$
$$|E_{att}|=1$$
$$|E_{noatt}|=9$$

$$m_{att}=min(1.0,\frac{AF\cdot n_{anhang,att}}{|E_{att}|}) \quad \Rightarrow 1.0$$

$$m_{noatt}=min(1.0,\frac{NF\cdot n_{anhang,noatt}}{|E_{noatt}|}) \quad \Rightarrow 0.33$$

AF,NF sind Gewichte für Wörter in den Klassen noatt, att (hier AF=1, NF=1)

$$m_{total}=max(NP,min(AP,\frac{m_{att}}{m_{att}+m_{noatt}})) \quad \Rightarrow 0.75$$

NP=Wert für Noattachment Wort
AP = Wert für ein Attachment Wort
(hier AP=0.99, AP=0.01)

m_{total} gibt die Wahrscheinlichkeit an, dass eine E-Mail ein Attachment enthält

Paul Graham Beispiel Ergebnis

TECHNISCHE UNIVERSITÄT DARMSTADT

Vorberechnetes Verhältnis

Anhang	0.75
Im	0.776
sind	0.209
Unterlagen	0.01
Bilder	0.99

Neues Beispiel: `Im Anhang sind Bilder`

Bilder	0.99
Im	0.776
Anhang	0.75
sind	0.209

$$p = 0.99 \cdot 0.776 \cdot 0.75 \cdot 0.209 = 0.12$$ **=> 99.63 %**

$$ip = (1-0.99) \cdot (1-0.776) \cdot (1-0.75) \cdot (1-0.209) = 0.00044$$

Paul Graham versus Naive Bayes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Beide Verfahren gute Erkennungsraten
- Naive Bayes hat beim Klassifizieren mehr zu berechnen
- Paul Graham bezieht nur die ersten 15 Wörter ein
- Paul Graham setzt Anzahl der Wörter und Anzahl der E-Mails in Relation

Evaluierung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mit welchem Algorithmus lässt sich das Problem gut lösen ?
- Wie groß muss die zu lernende Menge an E-Mails sein ?
- Wieviele Fehler macht der Algorithmus ?

Testmengen

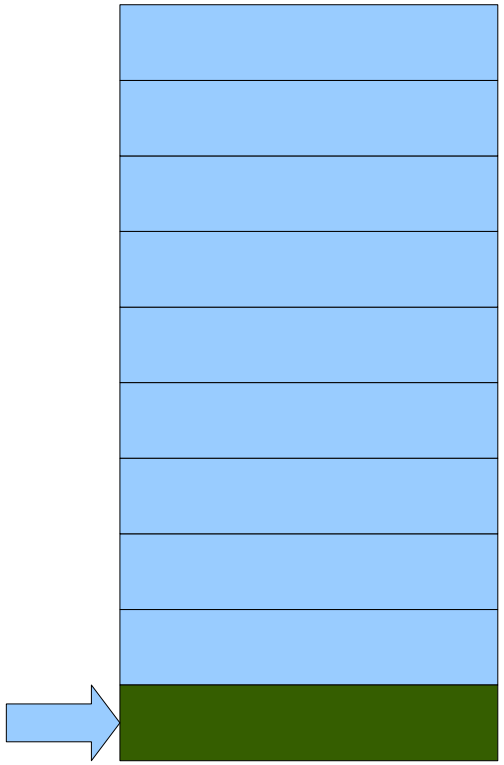
TECHNISCHE UNIVERSITÄT DARMSTADT

- Vier Testmengen
  - 1. 2180 deutsche E-Mails (21,33% Att.)
  - 2. 1406 deutsche E-Mails (11,81% Att.)
  - 3. 690 deutsche E-Mails (21,30% Att.)
  - 4. 420 deutsche E-Mails (11,81% Att.)

Cross-Validation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kein einzelnes Testset verfügbar
- Teilen der bestehenden E-Mails
- 9 Teile zum Lernen und 1 Teil zum Test
- 10 Durchgänge



Einige Definitionen

TECHNISCHE UNIVERSITÄT DARMSTADT

	Vorhergesagtes Attachment	Vorhergesagtes Noattachment
Attachment	true positive	false negative
Noattachment	false positive	true negative

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}$$

Accuracy

TECHNISCHE UNIVERSITÄT DARMSTADT

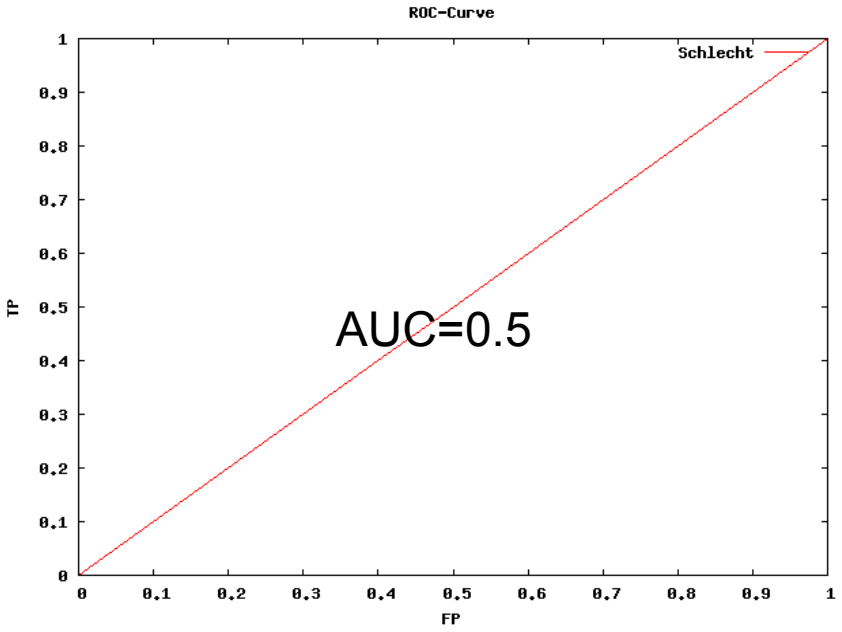
Accuracy	Testset 1	Testset 2	Testset 3	Testset 4
Einfacher Alg.*	76.01%	-	77,25%	81,43%
Erweiterter Alg.	73,72% (15 - 78,44%)	36,56% (15 - 70,06%)	75,94% (15 - 83,19%)	84,76% (15 - 88,33%)
Naive Bayes 0.55	78.81%	56,05% (0.60 - 92.60%)	73,67%	88,10%
Paul Graham 0.01	**81,97%**	**91,96%**	77,00% (0.4 - **82.17%**)	**89,52%**
Immer Att.	21,33%	11,81%	21,30%	11,43
Nie Att.	78,67%	88,19%	78,70%	88,57%

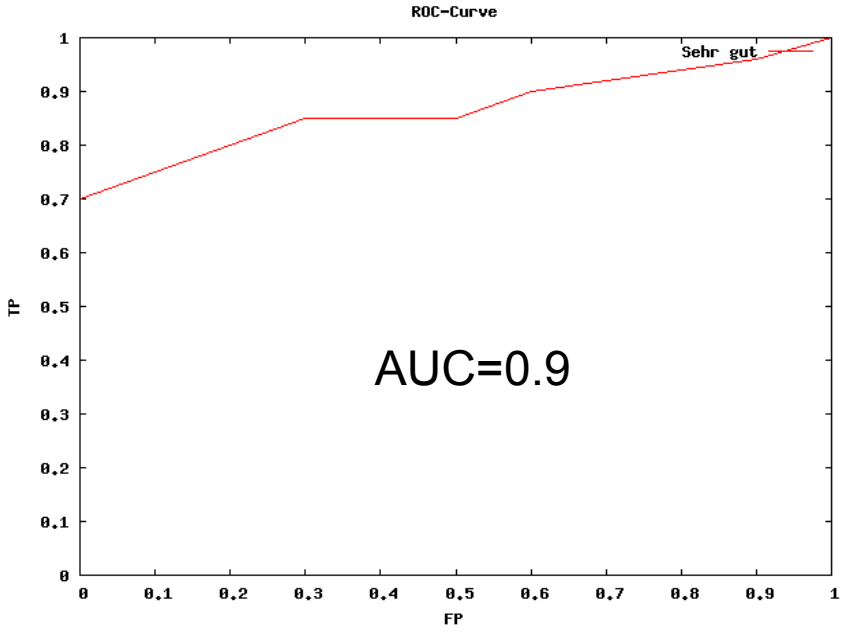
*words=['anhang','anhänge','datei','dateien','fotos','version','versionen','entwurf','entwürfe','anbei']

ROC-Kurve

TECHNISCHE UNIVERSITÄT DARMSTADT

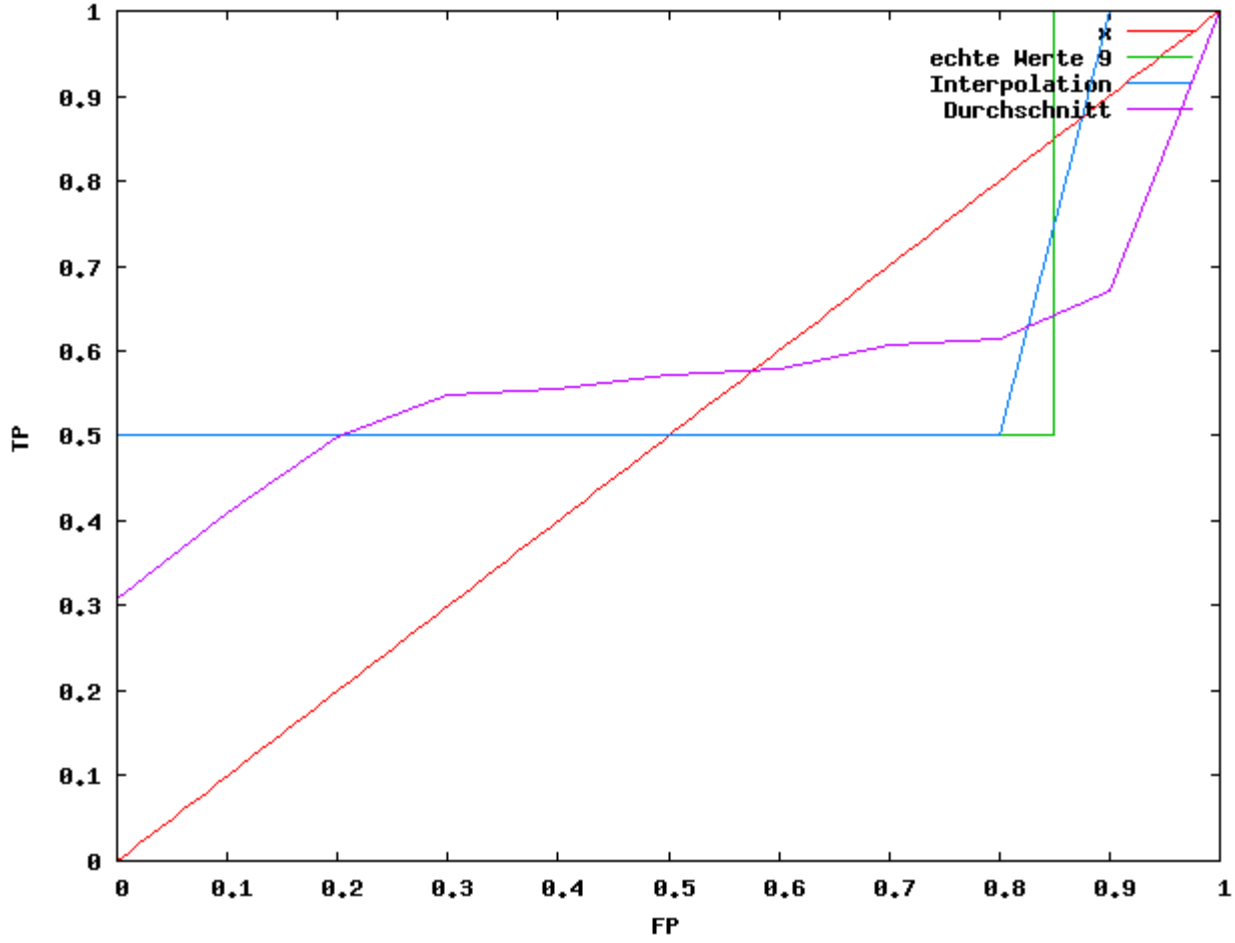
- ROC-Kurve stellt Trennbarkeit der Klassen dar
- Schwellwert wird nicht betrachtet
- E-Mails absteigend nach Wert/Wahrscheinlichkeit sortiert (0.99, 0.98...)





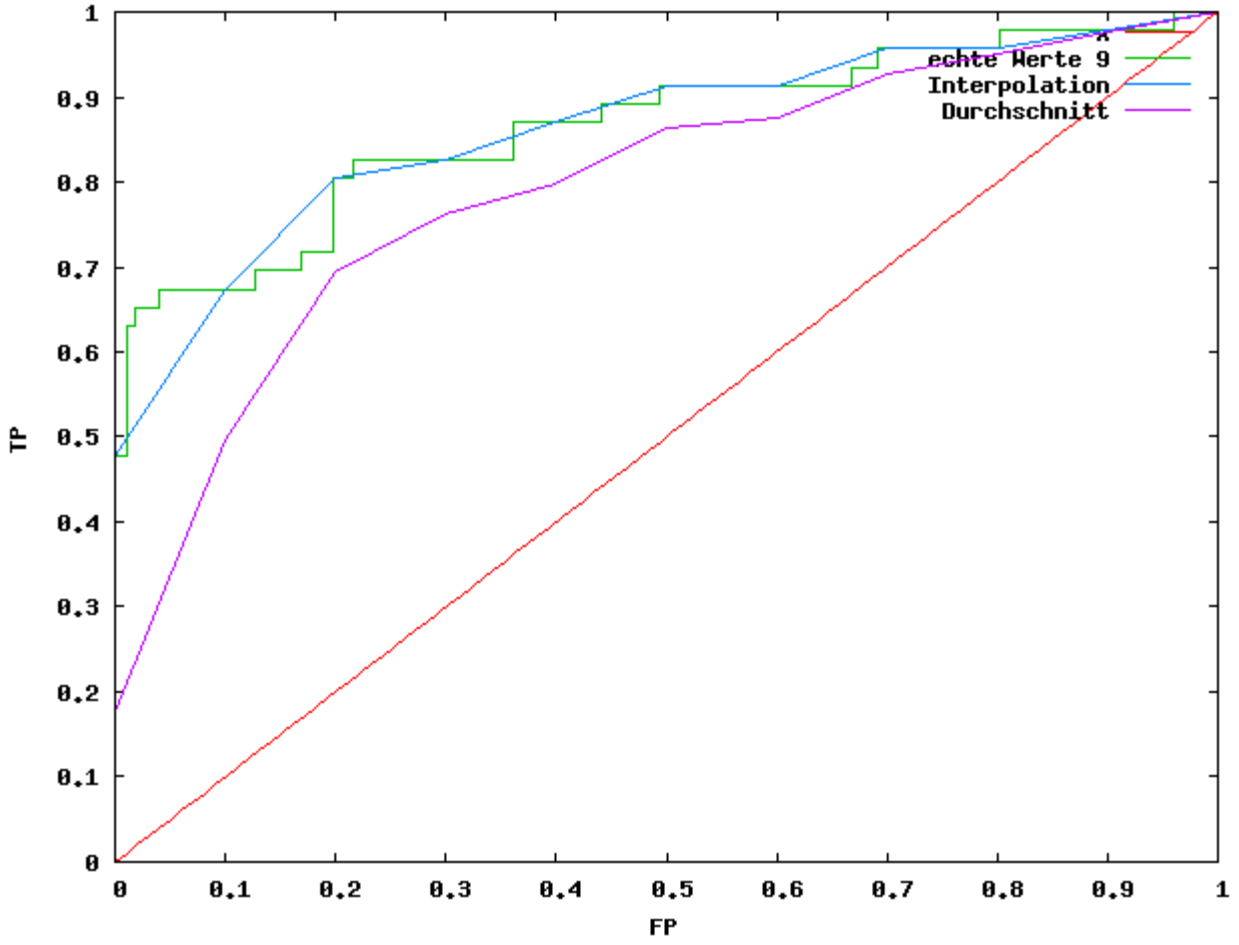
ROC-Kurve kleiner Datensatz

TECHNISCHE UNIVERSITÄT DARMSTADT



ROC-Kurve großer Datensatz

TECHNISCHE UNIVERSITÄT DARMSTADT



Area under curve

TECHNISCHE UNIVERSITÄT DARMSTADT

	Testset 1	Testset 2	Testset 3	Testset 4
Einfacher Alg.	0.58	-	0.67	0.65
Erweiterter Alg.	0.52	0.45	0.69	0.50
Naive Bayes	0.65	0.70	0.60	0.76
Paul Graham	0.80	0.86	0.74	0.79

AUC = 1.0 alle positiven Beispiele liegen über den negativen
AUC = 0.5 zufällig gewählt
AUC = 0.0 alle negativen Beispiele liegen über den positiven

Fazit

TECHNISCHE UNIVERSITÄT DARMSTADT

- Es funktioniert...
- Noch viele Fehlalarme
- Spam-Problem ist besser lösbar

Ausblick

TECHNISCHE UNIVERSITÄT DARMSTADT

- Forschungskorpus
- E-Mails besser zerlegen
- Plugin öffentlichkeitstauglich machen (0.32)
- Verfügbarmachen für andere Plattformen

Fragen ?

TECHNISCHE UNIVERSITÄT DARMSTADT

Vielen Dank für Ihre Aufmerksamkeit !

Anhang

TECHNISCHE UNIVERSITÄT DARMSTADT

- Worttrennung
- Implementierung
- ROC - Erklärung

Worttrennung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Trennung nach [äÄüÜöÖa-zA-Z0-9'_-]+
- Möglicher Optimierungspunkt
- Jeder Algorithmus benutzt diese Trennung
- Wortlängenbegrenzung sinnvoll

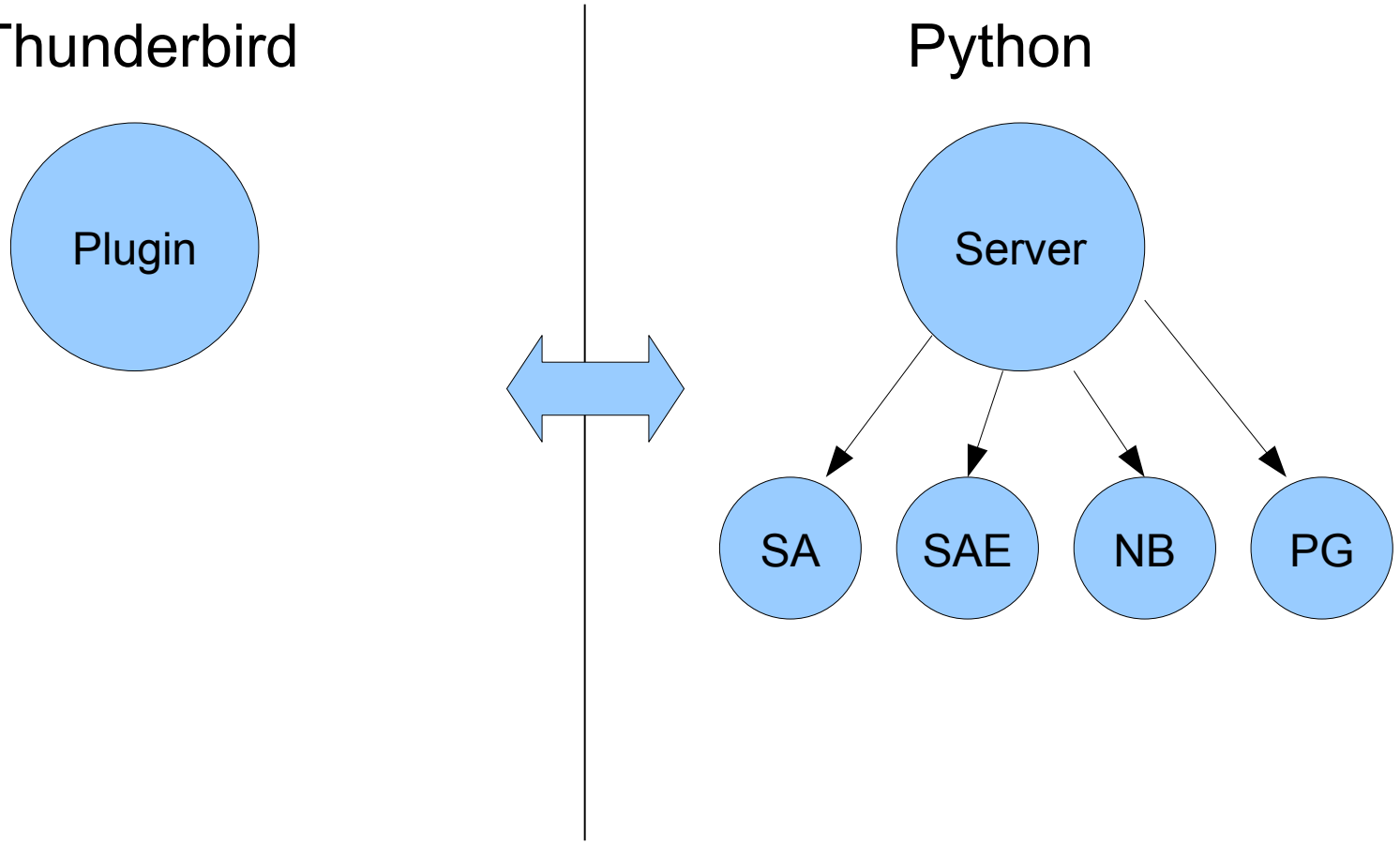
Implementierung
Einleitung

TECHNISCHE UNIVERSITÄT DARMSTADT

- Kurze informelle Erklärung
- Thunderbird-Plugin als Client
- Python als Server
  - Testbarkeit
  - Standardbibliotheken

Aufbau Plugin

TECHNISCHE UNIVERSITÄT DARMSTADT



Ablauf E-Mail-Versand

TECHNISCHE UNIVERSITÄT DARMSTADT

Thunderbird

1. Nachricht verfassen
2. Nachricht an Server

4. Antwort verarbeiten
Ja und Att => Senden
Ja und Noatt => Popup
Nein und Att => Senden
Nein und Noatt => Senden

5. Endgültige Nachricht schicken

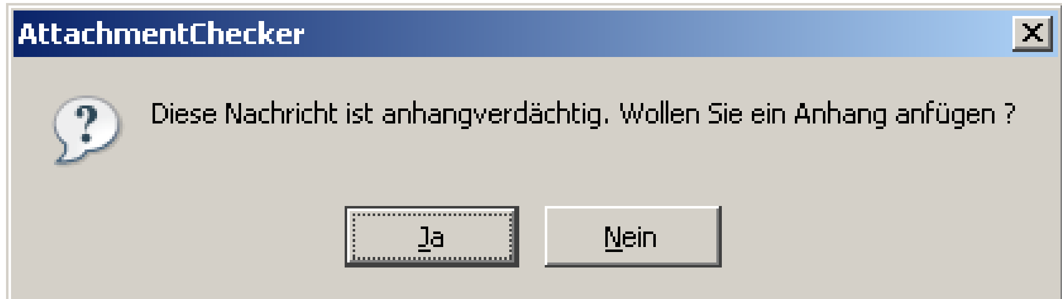
Python

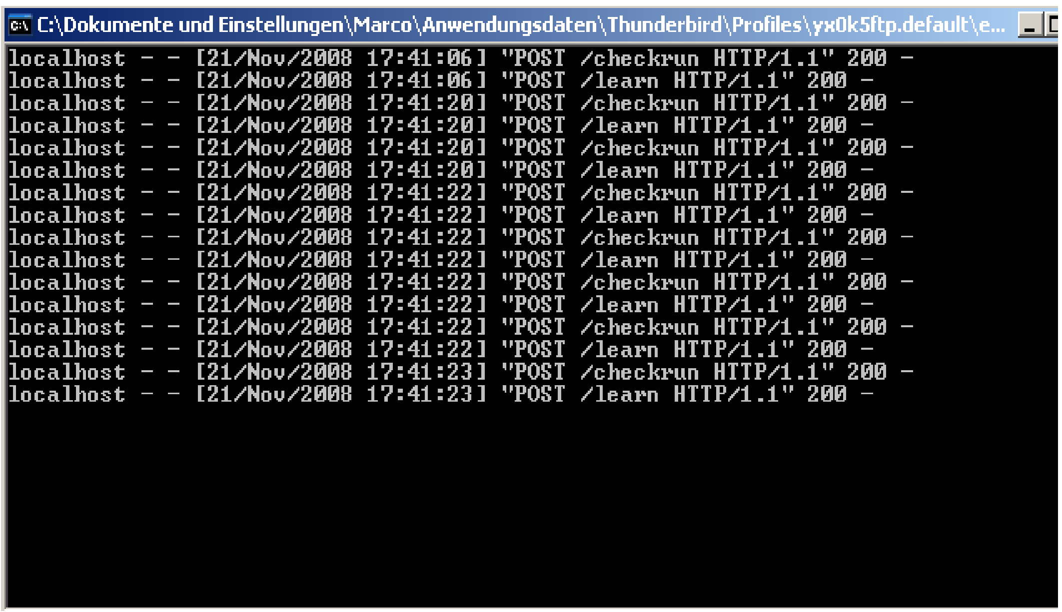
3. Nachricht prüfen und abhängig vom Algorithmus Ergebnis schicken (Ja/Nein)

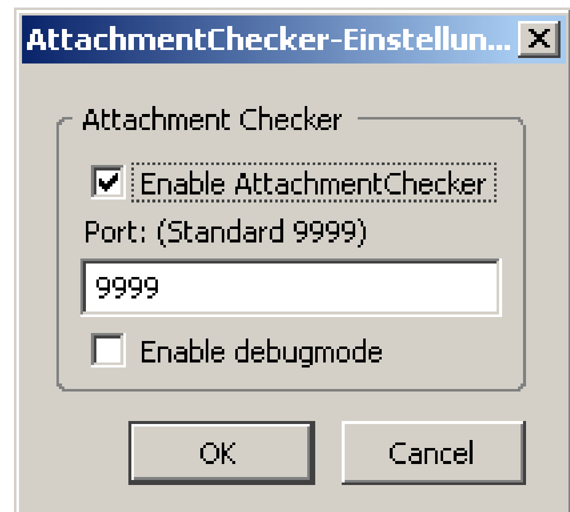
6. Nachricht verarbeiten und ggfls. Lernen

Screenshots

TECHNISCHE UNIVERSITÄT DARMSTADT



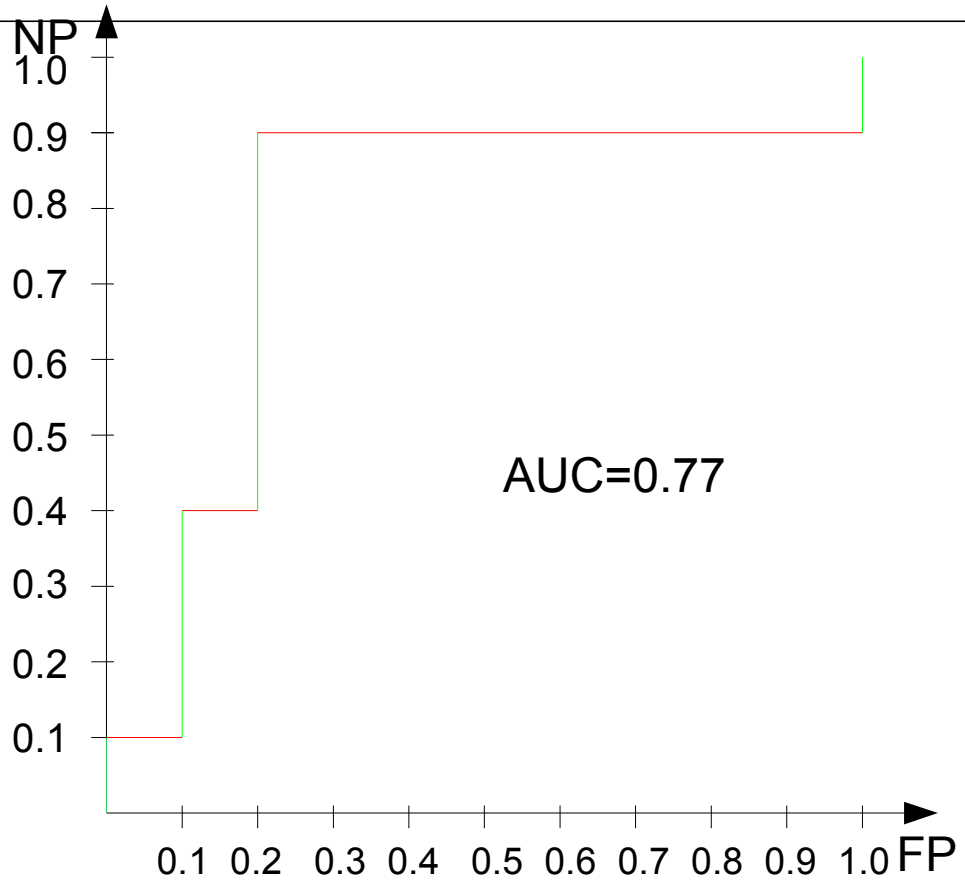




ROC Erklärung

TECHNISCHE UNIVERSITÄT DARMSTADT

1. T 0.99	11.T 0.10
2. F 0.98	12.F 0.09
3. T 0.97	13.F 0.09
4. T 0.97	14.F 0.09
5. T 0.97	15.F 0.09
6. F 0.50	16.F 0.03
7. T 0.49	17.F 0.01
8. T 0.30	18.F 0.01
9. T 0.29	19.F 0.01
10.T 0.28	20.T 0.01



N = 10
P = 10

$$AUC = \frac{S - N\frac{(N+1)}{2}}{N \cdot P} = \frac{132 - 10 \cdot 5.5}{100} = 0.77$$