

# Übungsblatt

## Grundlagen der Künstlichen Intelligenz

10.11.2023, DHBW Lörrach

### Naive-Bayes Spam-Filter

#### Aufgabe 1

Wir wollen die Wahrscheinlichkeit dafür berechnen, dass eine empfangene E-Mail  $m$  entweder *Spam* oder *Ham* (erwünschte E-Mail) ist, unter der Bedingung, dass die Wörter  $m = (Wort_1, \dots, Wort_n)$  in der Mail gefunden wurden:

$$P(S|Wort_1 \wedge \dots \wedge Wort_n) = \frac{P(Wort_1 \wedge \dots \wedge Wort_n|S) \cdot P(S)}{P(Wort_1 \wedge \dots \wedge Wort_n)} \quad (1)$$

$$= \frac{P(Wort_1|S) \cdot \dots \cdot P(Wort_n|S) \cdot P(S)}{P(Wort_1) \cdot \dots \cdot P(Wort_n)} \quad (2)$$

a) Begründen Sie anhand der Gleichung die Naivität des Spam-Filters.

b) Welcher Klassifikations-Algorithmus beschreibt das Kategorisierungsverhalten des Spam-Filters ( $cat \in \{S, H\}$ )? Begründen Sie Ihre Auswahl.

	$classify(m) = classify(Wort_1, \dots, Wort_n)$
<input type="checkbox"/>	$= \underset{cat}{argmin} P(cat) \cdot \prod_{i=1}^n P(Wort_i cat)$
<input type="checkbox"/>	$= \underset{cat}{argmax} P(cat) \cdot \prod_{i=1}^n P(Wort_i cat)$
<input type="checkbox"/>	$= \underset{cat}{arg} P(cat) \cdot \prod_{i=1}^n P(Wort_i cat)$

#### Aufgabe 2

Betrachten Sie das Python Code-Fragment aus einer Zeitreihenanalyse von Passagierzahlen zwischen 1949 und 1960, in dem die Zeitreihe mit dem Dickey-Fuller Test auf Stationarität geprüft wird:

```
from statsmodels.tsa.stattools import adfuller
def test_stationarity(timeseries):

    #Determing rolling statistics
    rolmean = pd.rolling_mean(timeseries, window=12)
    rolstd = pd.rolling_std(timeseries, window=12)

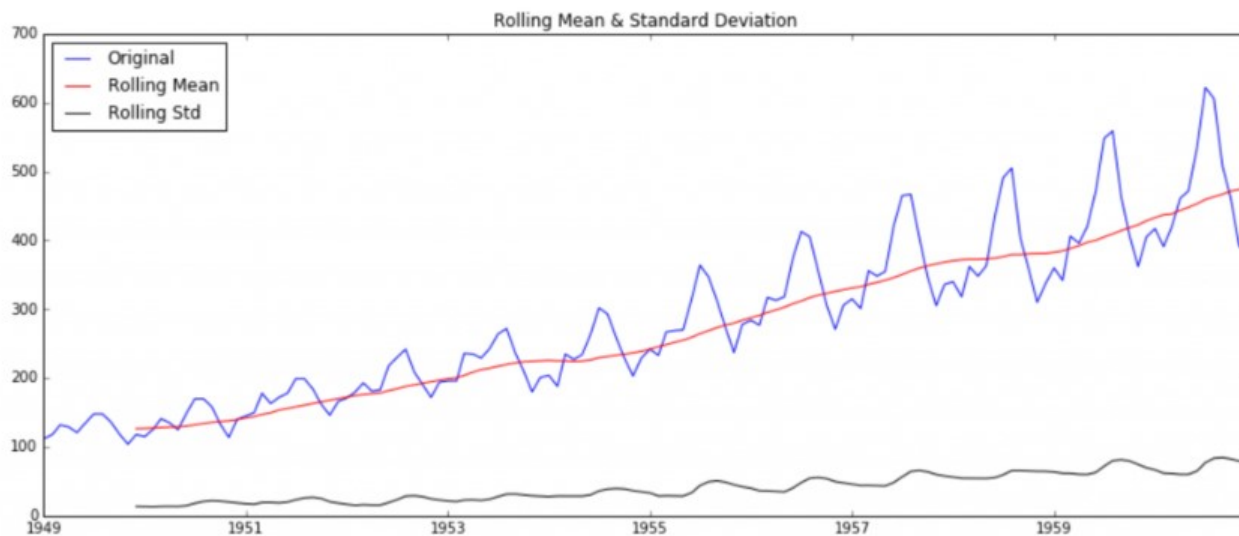
    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    #Perform Dickey-Fuller test:
    print 'Results of Dickey-Fuller Test:'
    dfctest = adfuller(timeseries, autolag='AIC')
    dfcoutput = pd.Series(dfctest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])
    for key,value in dfcoutput[4:].items():
        dfcoutput['Critical Value (%s)'%key] = value
    print dfcoutput
```

a) Formulieren Sie die Nullhypothese zur Prüfung auf Stationarität im Dickey-Fuller Test.

b) Welche Aussage über die Nullhypothese lässt sich entsprechend der zeitlichen Abhängigkeit des Erwartungswertes basierend auf dem Plot ableiten?

```
test_stationarity(ts)
```



Results of Dickey-Fuller Test:

Test Statistic	0.815369
p-value	0.991880
#Lags Used	13.000000
Number of Observations Used	130.000000
Critical Value (5%)	-2.884042
Critical Value (1%)	-3.481682
Critical Value (10%)	-2.578770
dtype:	float64

c) Recherchieren Sie - z.B. online - Kriterien, die für die Widerlegung der Null-Hypothese aus a) erfüllt sein müssen und geben die entsprechende Quelle an.