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 $m = (Wort_1, ..., Wort_n)$ in der Mail gefunden wurden:

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

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

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

Lösung

Für die zweite Gleichung wurde die Annahme verwendet, dass die gefundenen Wörter voneinander stochastisch unabhängig auftreten (Gegenbeispiele: *Baden* und *Württemberg*, *Naiv* und *Bayes*).

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

$$classify(m) = classify(Wort_1, ..., Wort_n)$$

☐ $$= \underset{cat}{argmin}\ P(cat) \cdot \Pi_{i=1}^{n} P(Wort_i|cat)$$

☐ $$= \underset{cat}{argmax}\ P(cat) \cdot \Pi_{i=1}^{n} P(Wort_i|cat)$$

☐ $$= \underset{cat}{arg}\ P(cat) \cdot \Pi_{i=1}^{n} P(Wort_i|cat)$$

Lösung

- zweite Möglichkeit
- Begründung: Die Auswahl fällt durch direkten Vergleich mit der Formel auf diejenige Kategorie, die die bedingte Wahrscheinlichkeit maximiert.

Aufgabe 2

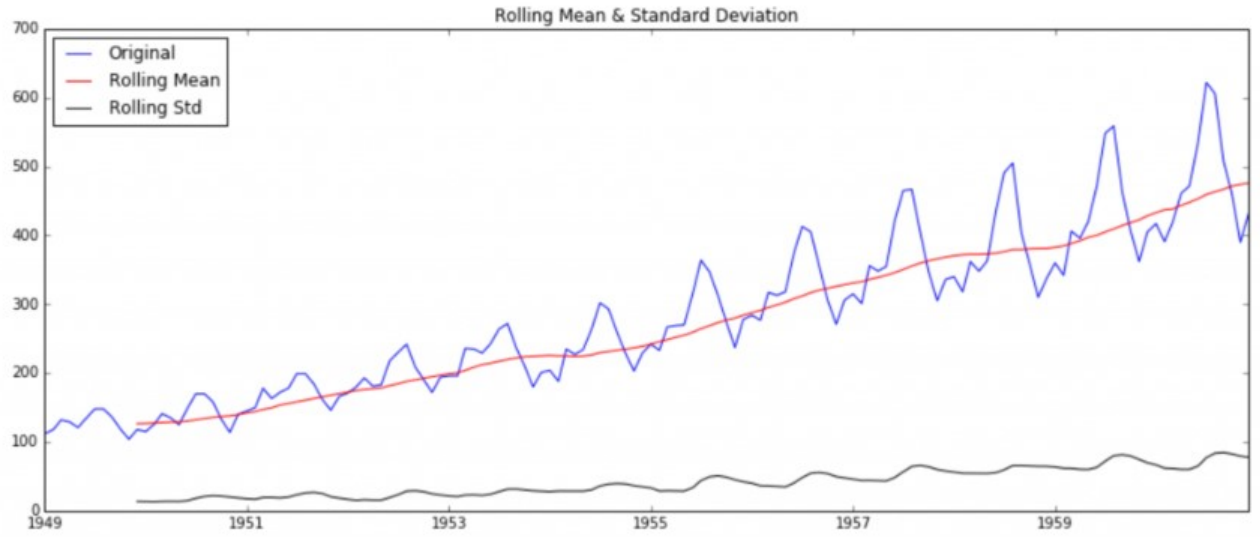
Betrachten Sie das Python Code-Fragment aus einer Zeitreihenanalyse von Passagierzahlen zwischen 1949 und 1960, in dem die Zeitreihe mit dem Dickey-Fuller Test auf Stationarität geprüft wird:

```
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:'
    dftest = adfuller(timeseries, autolag='AIC')
    dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observa
    for key,value in dftest[4].items():
        dfoutput['Critical Value (%s)'%key] = value
    print dfoutput
```

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
```

Lösung

Der Erwartungswert ist nicht konstant, sondern monoton steigend mit der Zeit, was die Stationarität der Zeitreihe widerlegt.

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.

Lösung

- Test Statistic (ca. 0.82) < Critical Value (ca. -2.6 bei 10 Prozent Konfidenzintervall) und
- p-value (vgl. MacKinnon) < 0.05
- Quelle: https://www.analyticsvidhya.com/blog/2021/06/statistical-tests-to-check-stationarity-in-time-series-part-1/