

Klausur zu Healthcare Data Analytics

Dr. Michael Strobel

15.07.20022

Organisatorisches

Vorname:

Nachname:

Matrikelnummer:

- Die Klausur dauert 90 Minuten.
- Erlaubte Hilfsmittel: ein doppelseitig beschriebenes Din-A4 Blatt und ein (nicht programmierbarer) Taschenrechner.
- Weitere Hilfsmittel sind nicht erlaubt.

Aufgabe 1: Allgemeines Verständnis

1. Was ist der Unterschied zwischen Trainings Set und Test Set?
2. Wofür dienen Label bei Supervised Learning?
3. Nennen Sie ein Beispiel für ein Unsupervised Learning.
4. Wie viel Prozent der Daten sollten in etwa für das Testset beiseite gelegt werden?
5. Was ist der Unterschied zwischen Klassifikation und Regression?
6. Was verstehen wir unter Underfitting?
7. Was was verstehen wir unter Overfitting?
8. Was verstehen wir unter Regularisierung eines Modells?
9. Was ist Semi-Supervised Learning?
10. Was ist ein Data Warehouse?

Aufgabe 2: Python Basics

a) Python Pakete

In der Vorlesung haben wir zahlreiche Software Pakete kennengelernt. Beschreiben Sie kurz die **Verwendungen** der unten stehen Pakete in der Vorlesung an. Desweiteren interpretieren Sie die **Kommandos** und deren **Output**.

Numpy

Code Beispiel

```
>>> import numpy as np
>>> age_in_years = np.array([10, 13, 18, 7, 19, 33, 45, 81, 60])
>>> np.where(age_in_years < 20, age_in_years, 0)
array([10, 13, 18,  7, 19,  0,  0,  0,  0])
```

Verwendung von Numpy:

NumPy ist eine Bibliothek, die eine einfache Handhabung von Vektoren, Matrizen oder generell großen mehrdimensionalen Arrays ermöglicht. Neben den Datenstrukturen bietet NumPy auch effizient implementierte Funktionen für numerische Berechnungen an.

Interpretation des Code Beispiels: Zuerst wird das numpy array `age_in_years` angelegt und mit `np.where` alle Einträge selektiert die kleiner als 20 sind. Andere werden auf 0 gesetzt.

Pandas

Code Beispiel

```
>>> import numpy as np
>>> import pandas as pd

>>> df = pd.DataFrame(np.arange(20).reshape(5,4), columns=["A","B","C","D"])
>>> df.loc[[0,1], "B"]
0    1
1    5
Name: B, dtype: int32
```

Verwendung von Pandas:

Pandas ist ein Paket zur Datenanalyse.

Es bietet insbesondere Datenstrukturen und Operationen für die Bearbeitung von numerischen Tabellen und Zeitreihen.

Interpretation des Code Beispiels:

b) Quicksort

Implementierung Sie Quicksort in Python 3

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)
```

```
print(quicksort([3, 6, 8, 10, 1, 2, 1]))
>>> print(quicksort([3, 6, 8, 10, 1, 2, 1]))
[1, 1, 2, 3, 6, 8, 10]
```

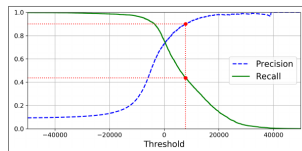
Hinweis: der rekursive Quicksort Algorithmus in Haskell kann folgendermaßen implementiert werden

```
quicksort [] = [] -- Die leere liste ist bereits sortiert
-- x:xs nimmt das erste Element aus der Liste und nennt es x, xs ist dann der Rest der Liste
quicksort (x:xs) = quicksort [a | a <- xs, a < x] -- Sortiere die linke Seite
                  ++ [x] ++ -- Sortiere x in die Mitte
                  quicksort [a | a <- xs, a >= x] -- Sortiere die rechte Seite
```

Aufgabe 3: Precision und Recall

a) Precision / Recall Tradeoff

1. Wie ist Precision definiert? Wie kann Precision interpretiert werden?
2. Wie ist Recall definiert? Wie kann Recall interpretiert werden?
3. Was versteht man unter dem Precision- und Recall Tradeoff? Interpretieren Sie hierfür das Schaubild aus der Vorlesung.



4. Sie haben einen Binären Klassifikator trainiert, der 1 ausgibt wenn eine Person krank ist und 0 wenn die Person gesund ist. In $y_{predicted}$ finden Sie die Vorhersagen des Algorithmus und in y_{true} finden Sie den korrekten Wert.

$y_{predicted} = [0, 1, 1, 0, 1, 1]$

$y_{true} = [0, 1, 0, 0, 1, 0]$

Berechnen Sie folgende Werte

- True Positive (TP)
- False Positive (FP)
- False Negative (FN)
- Precision
- Recall

Aufgabe 4: Pipelines

a) Verständnis

1. Was verstehen wir unter einer Data Pipeline?
2. Was ist der Unterschied zwischen einer Kategorischen und einer Numerischen Pipeline?
3. Nennen Sie 3 Schritte die Sie in einer Numerischen Data Pipeline ausführen sollten. Begründen Sie Ihre Antwort.
4. Nennen Sie 3 Schritte die Sie in einer Kategorischen Data Pipeline ausführen sollten. Begründen Sie Ihre Antwort.

b) Pipeline in Python

Gegeben seien folgen Daten die wir in einen Pandas Dataframe mit dem Variablennamen `heart_data` laden:

	Age	Sex	ExerciseAnges	RestingBloodPressure	Cholesterol	MaxHeartRate	ExerciseAnges	RestingBloodPressure	Cholesterol	MaxHeartRate
0	39	M	1	101	263	155	Normal	150	240	150
1	41	F	1	102	263	155	Normal	150	240	150
2	43	M	1	103	263	155	Normal	150	240	150
3	45	M	1	104	263	155	Normal	150	240	150
4	47	M	1	105	263	155	Normal	150	240	150

Ergänzen Sie den unten stehenden Python 3 Code um sowohl eine kategorische als auch eine numerische Pipeline zu bauen und diese dann zu einer Gesamtpipeline zusammenzusetzen.

```
import numpy as np
import pandas as pd
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import (
    LabelEncoder,
    OneHotEncoder,
    OrdinalEncoder,
    StandardScaler,
)
from sklearn.impute import SimpleImputer

heart_data = pd.read_csv("data/heart_unclean.csv")

ordinal_features =
nominal_features =
numeric_features =

numeric_transformer = Pipeline(
    steps=[

    ]
)
nominal_transformer = Pipeline(
    steps=[
```

```

        ]
    )
    ordinal_transformer = Pipeline(
        steps=[

            ]
    )

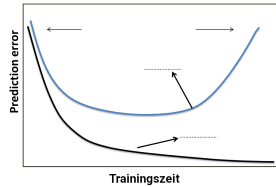
    full_pipeline = ColumnTransformer(
        transformers=[

            ]
    )

```


Aufgabe 5: Neuronale Netze

a) Verständnis



- Welche der Kurven ist wahrscheinlicher der Trainingsfehler und welche ist der Validierungsfehler? Geben Sie dies im Diagramm an, indem Sie die gepunkteten Linien ausfüllen.
- In welchen Bereichen des Diagramms sind Bias und Varianz niedrig bzw. hoch? Geben Sie deutlich an auf dem Diagramm mit vier Beschriftungen: “geringe Varianz”, “hohe Varianz”, “geringer bias”, “hoher bias”.
- In welchen Bereich underfitted bzw. overfitted das Modell? Geben Sie dies im Diagramm mit “Overfitting” und “Underfitting” an.

b) AlexNet

Auf dem Bild ist die Architektur von AlexNet zu sehen. Ergänzen Sie den unten stehenden TensorFlow Code um AlexNet in Keras zu implementieren.

Layer	Type	Maps	Size	Kernel size	Stride	Padding	Activation
Out	Fully connected	—	1,000	—	—	—	Softmax
F10	Fully connected	—	4,096	—	—	—	ReLU
F9	Fully connected	—	4,096	—	—	—	ReLU
S8	Max pooling	256	6×6	3×3	2	valid	—
C7	Convolution	256	13×13	3×3	1	same	ReLU
C6	Convolution	384	13×13	3×3	1	same	ReLU
C5	Convolution	384	13×13	3×3	1	same	ReLU
S4	Max pooling	256	13×13	3×3	2	valid	—
C3	Convolution	256	27×27	5×5	1	same	ReLU
S2	Max pooling	96	27×27	3×3	2	valid	—
C1	Convolution	96	55×55	11×11	4	valid	ReLU
In	Input	3 (RGB)	227×227	—	—	—	—

```
from keras.layers import Conv2D, MaxPool2D, Flatten, Dense, Dropout
```

```
model = keras.models.Sequential([
    Conv2D(filters=128, kernel_size=(11,11), strides=(4,4), activation='relu', input_shape=(64,64,3)),
    MaxPool2D(pool_size=(2,2)),
```

```
Dense(10,activation='softmax')  
])
```

Aufgabe 6: Decision Tree und Random Forrest

1. Nennen Sie vier Eigenschaften von Decision Trees.
2. Was verstehen wir unter dem Gini Koeffizient? Wie interpretieren Sie Gini 0.1 und Gini 0.5?
3. Wie kann ein Decision Tree Regularisiert werden?
4. Erklären Sie das Konzept von Bagging und Boosting. Was ist der Unterschied?
5. Wie werden Decision Trees konstruiert?
6. Betrachten Sie das Schaubild A. Welches Bilder zeigt einen Decision Tree und welches ein Random Forrest? Begründen Sie Ihre Antwort.

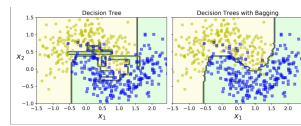


Figure 1: Schaubild A