Maschinelles Lernen

ML-Projekt

Hochschule Karlsruhe
University of
Applied Sciences

Prof. Dr. Patrick Baier

WS21/22

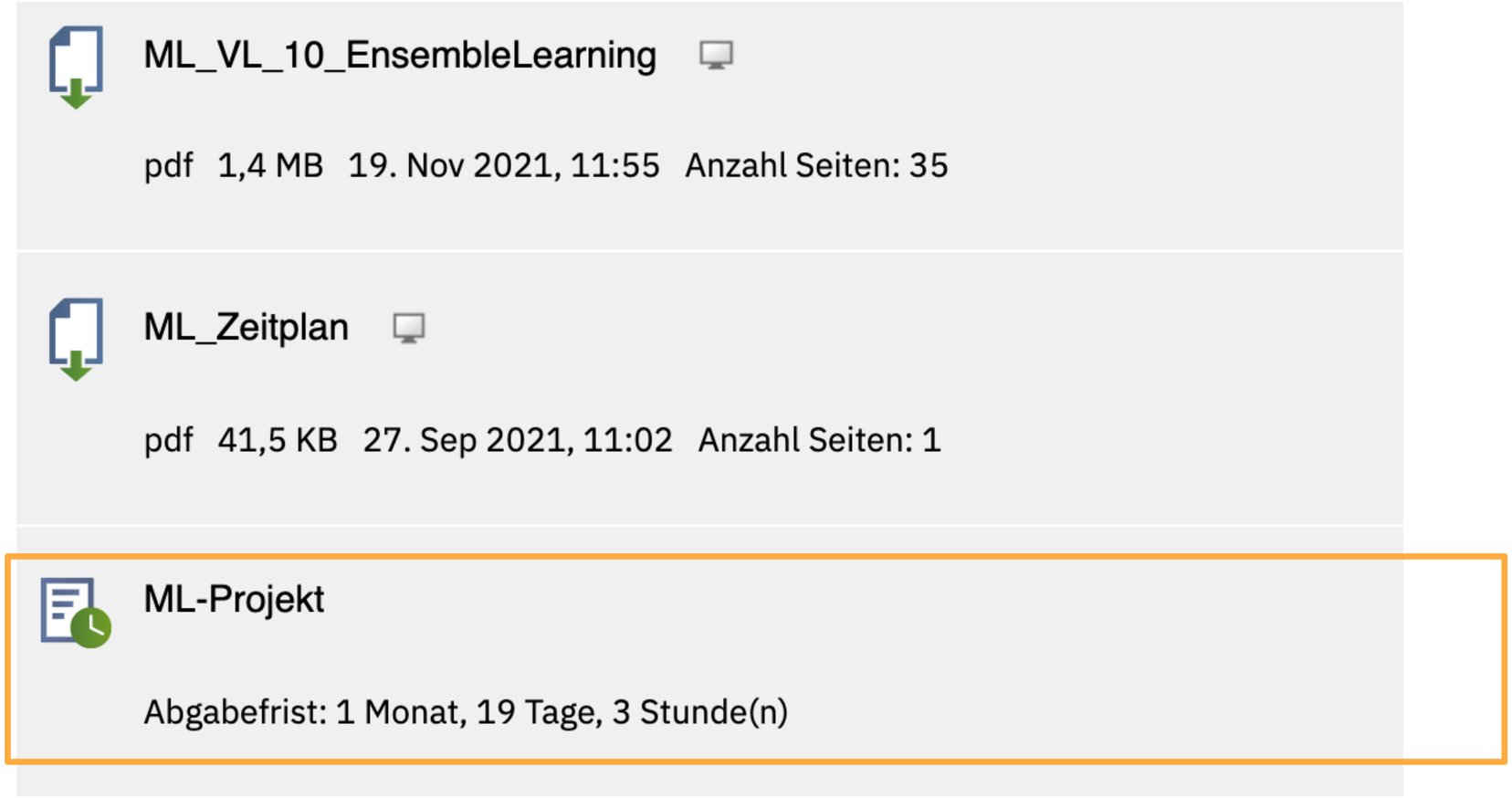
ML-Projekt

- Idee: Eigenständige Durchführung eines ML-Projekts auf einem gegebenen Datensatz in Teams mit je drei Personen.
- Zum Bestehen des Übungsscheins nötig:
  1. Abgabe des ML-Projekts als Jupyter-Notebook.
  2. 15 Minuten Code-Präsentation aller Teammitglieder.
- Zeitplan:
  - Herausgabe der Aufgabe: 30.11.
  - Abgabe der Notebook-Datei: 14.01.
  - Präsentationen: 18.01. und 25.01.

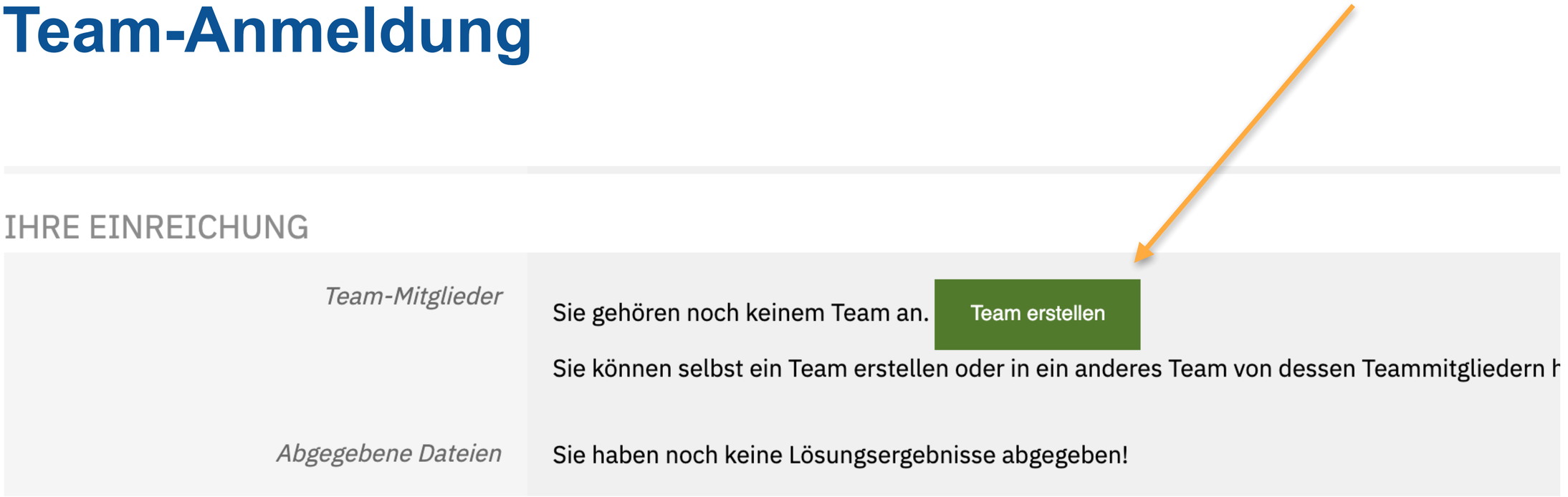
Teamfindung

- Die 3er-Teams können ab jetzt gesucht und gebildet werden.
- Teamsuche im Mattermost-Channel „Teams“.
- Ein Team kann auch aus Mitgliedern unterschiedlicher Übungsgruppen bestehen.
- Abnahmetermine am 18.01. und 25.01.:
  - Wenn alle Team-Mitglieder aus Ü-Gruppe 1: 11:30-13:00 oder 14:00-15:30
  - Wenn alle Team-Mitglieder aus Ü-Gruppe 2: 11:30-13:00 oder 15:40-17:10
  - Wenn gemischtes Team: 11:30-13:00 oder 14:00-14:30 oder 15:40-17:10
- Wer bis nächste Woche kein 3er-Team gefunden hat schreibt mir in Mattermost eine Nachricht, damit ich bei der Suche helfen kann.

Anmeldung und Abgabe



Team-Anmeldung



Achtung: Das Tool erlaubt beliebige Gruppengrößen, es sind aber nur 3er Teams erlaubt!

Präsentation

- Präsentationen am 18.01. und 25.01. zur Vorlesungs- und Übungszeit.
- Jedes Team erhält einen 15 Minuten Slot (mehr dazu später).
- Präsentation findet ggf. online statt, d.h. bis dahin sicherstellen, dass Kamera und Mikrofon funktionieren.
- Anwesende: Team, Bernardo und ich.
- Inhalt: Fragen über die Abgabe, z.B. „was macht dieser Befehl“, „warum brauchen wir diese Zeile“, „was bedeuten die Funktionsparameter“, …
- Jeder im Team muss über den ganzen Code Bescheid wissen. Antworten wie „das weiß ich nicht, das habe ich nicht programmiert“ oder „das weiß ich nicht, das haben wir einfach so von XY übernommen“, führen zum Nicht-Bestehen des Scheins für das jeweilige Team-Mitglied.

Datensatz

- Szenario: Wir betreiben einen Bücher Online-Shop und wollen beim Bestellvorgang vorhersagen ob ein Kunde seine Bestellung zurückschickt.
- Gegeben sind historische Bestelldaten in folgendem Format:

	transactionId	basket	customerType	totalAmount	returnLabel
0	7934161612	[3]	existing	77.0	0
1	5308629088	[5, 3, 0, 3]	existing	64.0	0
2	1951363325	[3, 3, 1, 4]	new	308.0	1
3	6713597713	[2]	existing	74.0	0
4	8352683669	[4, 4, 4, 4]	new	324.0	1

- Aufgabe: Bauen Sie ein ML-Modell welches das Label `returnLabel` vorhersagt.

Datensatz

	transactionId	basket	customerType	totalAmount	returnLabel
0	7934161612	[3]	existing	77.0	0
1	5308629088	[5, 3, 0, 3]	existing	64.0	0
2	1951363325	[3, 3, 1, 4]	new	308.0	1
3	6713597713	[2]	existing	74.0	0
4	8352683669	[4, 4, 4, 4]	new	324.0	1

- `transactionId` : Vom unserem Online-System zufällig generierte ID der Bestellung.
- `basket` : Welche Bücher wurden bestellt (siehe nächste Folie).
- `customerType` : Ist der Kunde neu oder hat er schon bei uns bestellt?
- `totalAmount` : Wie hoch ist der Warenwert der Bestellung?
- `returnLabel` : Wurde Artikel zurückgeschickt (= 1) oder behalten (= 0).

Datensatz

	transactionId	basket	customerType	totalAmount	returnLabel
0	7934161612	[3]	existing	77.0	0
1	5308629088	[5, 3, 0, 3]	existing	64.0	0
2	1951363325	[3, 3, 1, 4]	new	308.0	1
3	6713597713	[2]	existing	74.0	0
4	8352683669	[4, 4, 4, 4]	new	324.0	1

- Das Attribut `basket` enthält eine Liste von Bücherkategorien im Warenkorb.
- Die Werte im Basket entsprechen Bücherkategorien von 0 bis 5. (Zum Beispiel 0: Roman, 1: Science Fiction, 2: Gedichtband, usw.)
- Beispiel: `basket = [5, 3, 0, 3, 1, 1]`
  Der Kunde hat ein Buch der Kategorie „5“, zwei Bücher der Kategorie „3“, ein Buch der Kategorie „0“ und zwei Bücher der Kategorie „1“ bestellt. In der Bestellung ist kein Buch der Kategorie „2“ und „4“.

Datensatz

- Der Datensatz liegt in zwei Dateien vor: `train.csv` und `test.csv`
- Beide können über GitHub unter `data/project` runtergeladen werden.
- Die Daten in `train.csv` sollen für das Modelltraining und die Cross-Validation benutzt werden.
- Die Daten in `test.csv` werden zum Testen des Modells benutzt, um die finale Performance zu messen.
- Achtung: Das Trennzeichen („Delimiter“) in den Files ist ein Semikolon, das muss Pandas wissen, sonst schlägt das Datenladen fehl.

Aufgabe

- Allgemeines Ziel: Es soll ein Machine-Learning Modell trainiert werden, das eine möglichst hohe Accuracy auf den Testdaten erzielt.
  Aber: Die Testdaten dürfen während des Trainings nicht benutzt werden.
- Schritte:

  1. Laden Sie die Trainingsdaten.
  2. Führen Sie eine kurze EDA durch (Details auf Slide 14).
  3. Füllen Sie fehlende Werte in den Trainingsdaten auf.
  4. Transformieren Sie die kategorischen Features mittels One-hot-encoding.
  5. Versuchen Sie auf Basis des Attributs `basket` Features zu bauen (z.B. wie oft kommt jede Kategorie im Basket vor).

Aufgabe

6. Skalieren Sie die Features mit einem `StandardScaler`.
7. Trainieren Sie die folgenden Klassifikationsmodelle und probieren Sie die angegebenen Hyperparameter mittels Cross-Validation aus:
   1. Logistische Regression: *C :[0.1,1,4,5,6,10,30,100]* und *penalty: ["l1", "l2"]*
   2. Random Forest: *n_estimators: [60,80,100,120,140]* und *max_depth: [2, 4, 6]*
   3. Gradient Boosting Tree: gleiche Hyperparameter wie bei Random Forest.

Aufgabe

8. Laden Sie die Testdaten.
9. Entfernen Sie alle Zeilen mit fehlenden Werten.
10. Transformieren Sie die Attribute genauso wie bei den Trainingsdaten.
11. Skalieren Sie die Daten im gleichem Maß wie die Trainingsdaten.
12. Machen Sie eine Vorhersage auf den Testdaten mit allen drei Modellen und den jeweils besten Hyperparametern aus der Cross Validation.
13. Berechnen Sie für jedes der drei Modell Accuracy, Precision und Recall.
14. Berechnen Sie außerdem die Accuracy auf den Trainingsdaten und vergleichen Sie Accuracy auf Trainings- und Testdaten. Liegt Overfitting vor?

Aufgabe

15. Untersuchen Sie wieviele Datenpunkte es in den Testdaten gibt, welche von allen drei Modellen falsch klassifiziert wurden:
    1. Bestimmen Sie für jedes der drei Modelle die Indizes der Testdatenpunkte auf welchen das jeweilige Modell falsch klassifiziert hat.
    2. Nutzen Sie die `set`-Klasse in Python um die Anzahl an Datenpunkten zu bestimmen, welche von allen drei Modellen falsch klassifiziert wurden.

EDA (Exploratory Data Analysis)

- Nachdem Sie die Daten geladen haben, führen Sie kurz die folgenden Analysen durch:
  - Plotten Sie Histogramme zu den Features `customerType` und `totalAmount`
  - Ermitteln sie die Verteilung von `returnLabel` i) über alle Daten ii) in Abhängigkeit zu allen Ausprägungen von `customerType`.
  - Erstellen Sie einen Boxplot für `totalAmount` in Abhängigkeit des Labels (nutzen Sie dafür den Parameter `by=…`)
  - Installieren Sie das `Seaborn`-Package und erstellen Sie einen `displot` für `totalAmount` in Abhängigkeit des Labels, wie hier beschrieben. Versuchen Sie unterschiedliche Werte für das Argument `element`.

Hinweise

- Im Vergleich zu früheren Code-Beispielen sind die Trainings- und Testdaten in zwei verschiedenen Dateien. Das bedeutet, dass die Attribute für beide Dateien separat in Features überführt werden müssen.
- Mit folgender Methode können relativ einfach neue Features aus bestehenden Spalten erstellt werden (hilfreich für das `basket` Feature):

  z.B. `df["newFeature"] = df.basket.map(lambda x: len(x))`

  Dieser Befehl erzeugt eine neue Spalte welche die Länge des Baskets enthält (Tipp: Das Feature ist nicht sehr hilfreich und nur zur Demonstration gedacht).

Abgabe

- Eine Notebook-Datei im Format .ipynb + eine Textdatei (siehe nächste Seite)
- Das Notebook soll den angegebenen Schritten folgen und mit Markdown-Zellen kommentiert sein.
- Zum finalen Speichern des Notebooks, gehen Sie wie folgt vor:
  - Klicken Sie auf „Kernel -> Restart & Run All“
  - Warten Sie bis alle Zellen fehlerfrei ausgeführt wurden.
  - Klicken Sie auf das „Save“-Icon.
  - Klicken Sie dann auf „File -> Close and Halt“.
  - Das .ipynb-File des Notebooks ist jetzt bereit versendet zu werden.

Abgabe

- Die Textdatei ist leer und hat als Dateinamen die Nummer der Übungsgruppe in welcher die Teilnehmer sind. D.h. eine der Folgenden der Optionen:
  - Gruppe1.txt (d.h. alle Teilnehmer sind in Ü-Gruppe 1)
  - Gruppe2.txt (d.h. alle Teilnehmer sind in Ü-Gruppe 2)
  - Gruppe1u2.text (d.h. Teilnehmer sind sowohl in Ü-Gruppe 1 als auch Ü-Gruppe 2)
- Je nachdem in welcher Übungsgruppe die Teilnehmer sind findet ihre Abnahme in dem Slot der jeweiligen Übungsgruppe statt (oder im Vorlesungsslot).
- Wenn Teilnehmer in beiden Übungsgruppen, kommen beide Übungsslots in Frage.

Weiterer Ablauf

- Heute keine neuen Vorlesungsinhalte.
- Ab circa 13:00 Uhr sind wir (Oktavian, Bernardo, ich) im LAT und beantworten Fragen zu der Aufgabenstellung.

Fragen?