Analyse großer Datenmengen

Prüfungsleistung

12.09.2025

Allgemeines

- Der schriftliche Bericht muss bis 30. September 2025, 23:59 Uhr eingereicht werden.
- Deine Gesamtnote setzt sich zusammen aus:
  - 80 % schriftlicher Bericht
  - 20 % Code
- Die Daten können unter:
  https://myshare.leuphana.de/?t=d90d95a373674f928df5ff6c1999c054
  runtergeladen werden.
  Das Passwort ist: agd

Produkttext-Klassifikator

Stell dir vor, du arbeitest in einem schnell wachsenden E-Commerce-Startup – ähnlich den Anfängen von Amazon. Um das Einkaufserlebnis der Kund*innen zu verbessern, sollen Produktbeschreibungen automatisch passenden Kategorien zugeordnet werden. Du sollst dafür einen Proof of Concept (PoC) entwickeln, der als Entscheidungshilfe dient, ob sich eine solche Klassifizierung mit maschinellem Lernen realisieren lässt.

Bei der Entwicklung des Klassifikators müssen zahlreiche Entscheidungen getroffen werden. Orientiere dich bei der Entscheidung an den Anforderungen eines E-Commerce Shops. Bitte dokumentiere sowohl die aufgetretenen Probleme als auch deine Entscheidungen in deinem Bericht.

Beschreibe außerdem, welche nächsten Schritte nach dem PoC möglich wären – sowohl bezüglich der Daten als auch des Modells.

Daten

Der Datensatz besteht aus zwei Dateien:

- `prd_txt.parquet`
- `prd_typ.parquet`

Die Dateien enthalten folgende Spalten:
`prd_txt.parquet`:

- `produkt_id`: Produkt ID.
- `variation_id`: Variation innerhalb eines Produktes.
- `beschreibung`: Textuelle Beschreibung des Produkts.

`prd_type.parquet`:

- `produkt_id`: Produkt ID.
- `produkt_klasse`: Name der Produktklasse. **Diese soll vorhergesagt werden.**

Damit `parquet` Daten mit `pandas` verarbeitet werden können, muss noch das Python-Paket `pyarrow` installiert werden.

Worauf du achten solltest

- Es handelt sich um *echte Daten*. Das bedeutet, dass Daten fehlen, unvollständig oder inkonsistent sein können.
- Es ist nicht garantiert, dass sich mit diesen Daten ein guter Klassifikator erstellen lässt.
- Untersuche die Daten, bevor du mit der Analyse beginnst:
  - Welche Datentypen haben die einzelnen Spalten?
  - Enthalten die Spalten fehlende Werte? Falls ja, wie gehst du damit um?
  - Gibt es Extremwerte? Sind diese Artefakte?
  - Enthält das Textfeld unerwartete/unnötige Zeichen oder Strings?
  - Handelt es sich um eine Multi-Label-Aufgabe?
  - Gibt es Duplikate?
  - Gibt es ein Ungleichgewicht bei der Verteilung der Klassen?
    Welche Auswirkung kann das auf einen Embedder und Klassifiziere haben?
- Erstelle zunächst ein *Baseline-Modell*, also ein Modell, das sich sehr schnell berechnen lässt.
- Werden alle Daten für einen PoC benötigt?
- Gibt es Auffälligkeiten in der Konfusions-Matrix?

Schriftlicher Bericht

Du verfasst deinen Bericht für deine Kolleg*innen. Du kannst davon ausgehen, dass sie dasselbe technische Verständnis haben wie deine Kommiliton*innen. Sie kennen den Datensatz, haben ihn jedoch noch nicht analysiert.

Unter anderem interessiert sie:

- Welche Erkenntnisse du in den Daten gefunden hast.
- Vor welchen Entscheidungen du standest und wie du diese gelöst hast.
- Welche Werkzeuge und Methoden du verwendet hast.
- Wie gut der Klassifikator abschneidet.
- Welche Einschränkungen dein Ansatz / Embedder / Klassifikator hat.
- Was zu beachten ist, wenn das Modell produktiv eingesetzt wird.
- …

Code

Der von dir entwickelte Code fließt in die Bewertung ein. Fehlende Kommentare oder nicht aussagekräftig kommentierter Code führen zu Punkteabzug.

Vergiss nicht, den vollständigen Code mit einzureichen. Es können Jupyter-Notebooks als auch *.py Dateien eingereicht werden.

Die bereitgestellten Daten müssen nicht mit abgegeben werden.

Stelle sicher, dass dein Code lauffähig ist – er wird auf den bereitgestellten Daten getestet! Das heißt, der abgegebene Code muss von Anfang bis Ende durchlaufen. Zum Beispiel dürfen keine Daten geladen werden, die nicht im Code erstellt werden.

Falls Python Paket außer `jupyterlab, scikit-learn, matplotlib` verwendet werden, bitte als Kommentar angeben, wie die Pakete installiert wurden, idealerweise mit Version:
`pip install paket-name==1.2.3`