1. Einführung ([Try the Live Project Website]())

Ziel des Projektes ist es, meine Motivation für die ausgeschriebene Job Stelle aufzuzeigen und mich von anderen Bewerbern abzuheben.

- Dieser Abschnitt führt in die Problemstellung, Ziele und den Umfang des Empfehlungssystems für den E-Grocery-Store ein. Es wird das Ziel zusammengefasst, personalisierte Produktempfehlungen für Benutzer basierend auf deren bisherigen Käufen bereitzustellen.
- [Final Github repository]()

---

2. Datensatz kollection.

Quelle:
Die verwendeten Daten stammen aus dem Kaggle-Datensatz des Hunter's E-Grocery-Stores, der über 2 Millionen Kaufdatensätze enthält. [Dataset Link]().

Beschreibung des Datensatzes:

- Der Datensatz enthält Transaktionsinformationen wie Benutzer-IDs, Produkt-IDs, Bestelldetails, Wiederbestellstatus und Zeitstempel.
- Das Hauptziel ist es, ein Empfehlungssystem zu erstellen, das Produkte empfiehlt, die ein Benutzer wahrscheinlich kaufen wird, basierend auf vorherigen Produktauswahlen.

Sample:

	0	1	2	3	4
order_id	2425083	2425083	2425083	2425083	2425083
user_id	49125.0	49125.0	49125.0	49125.0	49125.0
order_number	1.0	1.0	1.0	1.0	1.0
order_dow	2.0	2.0	2.0	2.0	2.0
order_hour_of_day	18.0	18.0	18.0	18.0	18.0
days_since_prior_order	0	0	0	0	0
product_id	17.0	91.0	36.0	83.0	83.0
add_to_cart_order	1.0	2.0	3.0	4.0	5.0
reordered	0.0	0.0	0.0	0.0	0.0
department_id	13.0	16.0	16.0	4.0	4.0
department	pantry	dairy eggs	dairy eggs	produce	produce
product_name	baking ingredients	soy lactosefree	butter	fresh vegetables	fresh vegetables

3. Datenanalyse

3.1 Datenladen und Bereinigung:

Verwendete Bibliotheken:

- Pandas, Numpy für die Datenmanipulation.
- Matplotlib, Seaborn für Visualisierungen.

Bereinigungsmaßnahmen:

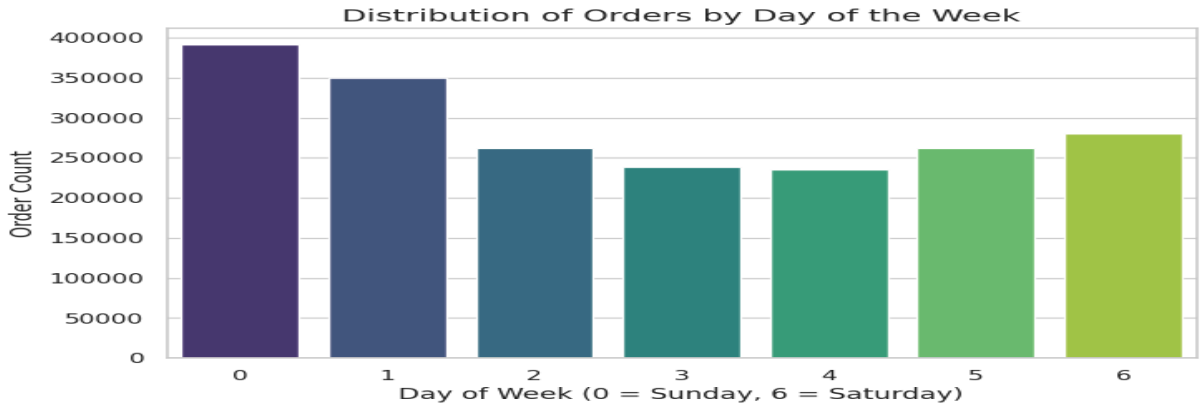
- Der Datensatz wird aus der Datei `/content/ECommerce_consumer_behaviour.csv` geladen.
- Fehlende Werte in der Spalte 'days_since_prior_order' werden mit 0 aufgefüllt und der Datentyp wird auf Integer geändert.
- Zeilen mit fehlenden Werten in der Spalte 'days_since_prior_order' werden entfernt.
- Doppelte Zeilen werden gelöscht.
- Es wird sichergestellt, dass die 'reordered'-Spalte nur 0 und 1 enthält.

3.2 Explorative Datenanalyse (EDA):

- **Grundlegende Statistiken:**
  Zusammenfassende Statistiken zu den numerischen Spalten wie Mittelwert, Median usw.
- **Visualisierungen:**
  - Verteilung der Bestellungen nach Wochentagen und Tagesstunden.
  - Verteilung der 10 meistbestellten Produkte und deren Wiederbestellhäufigkeit.
  - Heatmap zur Visualisierung der Korrelationsmatrix zwischen numerischen Merkmalen.

3.3 Datenexport:

Der bereinigte Datensatz wird in einer neuen Datei namens `e_commerce_data_Cleaned.csv` gespeichert.



4. Auswahl des Algorithmus

Das Empfehlungssystem verwendet zwei Ansätze:

- **Kollaboratives Filtern:**
  Gruppiert Produkte, die häufig zusammen gekauft werden, und empfiehlt diese, basierend auf der Kaufhistorie eines Benutzers.
- **Neural Network-basierte Empfehlung:**
  Neural Colloborative model wird trainiert, um das nächste Produkt vorherzusagen, das ein Benutzer kaufen könnte, basierend auf den Produkten in seiner aktuellen Bestellung und historischen Kaufdaten.

Nach der Bewertung beider Methoden wurde **Neural Network-basierte Empfehlung** aufgrund der besseren Ergebnisse gewählt.

---

5. Modelltraining

5.1 Datenladen und Vorverarbeitung:

- **Kodierung:**
  Benutzer- und Produkt-IDs werden in numerisches Format umgewandelt.
- **Datenvorbereitung:**
  Die Daten werden nach Benutzer und Bestellung sortiert, um die chronologische Reihenfolge der Käufe zu bewahren. Es werden Produktsequenzen für jeden Benutzer generiert, um Paare von aktuellen und nächsten Produkten zu erstellen.

5.2 Erstellung von Datensätzen und DataLoadern:

- Die Daten werden in Trainings-, Validierungs- und Testdatensätze unterteilt.
- Benutzerdefinierte Datensätze werden erstellt, um den Trainings-, Validierungs- und Testprozess zu verwalten.
- DataLoader werden verwendet, um die Batch-Verarbeitung effizient zu handhaben.

5.3 Architektur des Neuronalen Netzwerks:

- **Modelltyp:**
  Die Architektur basiert auf kollaborativem Filtern, bei dem das Modell das nächste Produkt basierend auf dem Benutzer und der vorherigen Produktgeschichte vorhersagt.
- **Schichten und Embeddings:**
  - Embeddings für Benutzer und Produkte werden erstellt.
  - Verborgene Schichten mit Dropout für Regularisierung.
  - Eine finale voll verbundene Schicht zur Produktvorhersage.
- **Aktivierungsfunktionen & Verlustfunktion:**
  ReLU wird für die Aktivierungen in den versteckten Schichten verwendet, und Kreuzentropie-Verlust wird für die Multi-Class-Klassifikation verwendet.

5.4 Trainingsprozess:

- Das Modell wird über mehrere Epochen trainiert, wobei der Trainings- und Validierungsverlust überwacht wird.
- Der Optimierer, der verwendet wird, ist Adam, und die Bewertungsmetriken umfassen Genauigkeit und Kreuzentropie-Verlust.
- Nach jeder Epoche wird die Leistung des Modells auf einem separaten Validierungsdatensatz überprüft.

5.5 Modell speichern:

Nach dem Training wird das Modell für zukünftige Inferenz gespeichert.

---

6. Modellauswertung

- **Bewertungsmetriken:**

```
Training fold 1
Epoch 1/5, Training Loss: 3.8636, Validation Loss: 3.7667, Validation Accuracy: 0.1688
Epoch 2/5, Training Loss: 3.8002, Validation Loss: 3.7385, Validation Accuracy: 0.1709
Epoch 3/5, Training Loss: 3.7650, Validation Loss: 3.7181, Validation Accuracy: 0.1722
Epoch 4/5, Training Loss: 3.7348, Validation Loss: 3.7047, Validation Accuracy: 0.1725
Epoch 5/5, Training Loss: 3.7079, Validation Loss: 3.6945, Validation Accuracy: 0.1735
Training fold 2
Epoch 1/5, Training Loss: 3.8652, Validation Loss: 3.7665, Validation Accuracy: 0.1686
Epoch 2/5, Training Loss: 3.8018, Validation Loss: 3.7396, Validation Accuracy: 0.1698
Epoch 3/5, Training Loss: 3.7683, Validation Loss: 3.7170, Validation Accuracy: 0.1713
Epoch 4/5, Training Loss: 3.7364, Validation Loss: 3.7054, Validation Accuracy: 0.1713
Epoch 5/5, Training Loss: 3.7108, Validation Loss: 3.6967, Validation Accuracy: 0.1717
Training fold 3
Epoch 1/5, Training Loss: 3.8647, Validation Loss: 3.7671, Validation Accuracy: 0.1685
Epoch 2/5, Training Loss: 3.8006, Validation Loss: 3.7386, Validation Accuracy: 0.1703
Epoch 3/5, Training Loss: 3.7659, Validation Loss: 3.7164, Validation Accuracy: 0.1724
Epoch 4/5, Training Loss: 3.7345, Validation Loss: 3.7039, Validation Accuracy: 0.1723
Epoch 5/5, Training Loss: 3.7080, Validation Loss: 3.6947, Validation Accuracy: 0.1729
Training fold 4
Epoch 1/5, Training Loss: 3.8620, Validation Loss: 3.7714, Validation Accuracy: 0.1687
Epoch 2/5, Training Loss: 3.7984, Validation Loss: 3.7432, Validation Accuracy: 0.1701
Epoch 3/5, Training Loss: 3.7630, Validation Loss: 3.7214, Validation Accuracy: 0.1718
Epoch 4/5, Training Loss: 3.7320, Validation Loss: 3.7080, Validation Accuracy: 0.1729
Epoch 5/5, Training Loss: 3.7065, Validation Loss: 3.6999, Validation Accuracy: 0.1728
Training fold 5
Epoch 1/5, Training Loss: 3.8619, Validation Loss: 3.7695, Validation Accuracy: 0.1683
Epoch 2/5, Training Loss: 3.7998, Validation Loss: 3.7429, Validation Accuracy: 0.1698
Epoch 3/5, Training Loss: 3.7648, Validation Loss: 3.7204, Validation Accuracy: 0.1715
Epoch 4/5, Training Loss: 3.7323, Validation Loss: 3.7068, Validation Accuracy: 0.1716
Epoch 5/5, Training Loss: 3.7063, Validation Loss: 3.6997, Validation Accuracy: 0.1711
Test Accuracy: 0.1711
```

- **Ergebnisse:**
  Das Neural Network-basierte Modell zeigte eine höhere Genauigkeit und bessere Generalisierung auf dem Testdatensatz im Vergleich zum kollaborativen Filtern.

---

7. Modellbereitstellung

7.1 Backend (Flask):

- **Flask-Framework:**
  Flask dient als Backend-API und verarbeitet Anfragen und liefert Produktempfehlungen basierend auf Benutzereingaben.
- **Endpunkte:**
  Das Backend stellt Endpunkte zur Verfügung, die mit dem trainierten Modell interagieren, um Produktempfehlungen abzurufen.

7.2 Frontend (ReactJS auf Vercel):

- **ReactJS:**
  Ein React-Frontend ermöglicht den Benutzern, mit dem Empfehlungssystem zu interagieren. Es sendet Produktdaten an das Flask-Backend und zeigt empfohlene Produkte den Benutzern an.

7.3 Modell-Hosting (Azure):

- Das trainierte Modell wird in einem Docker-Container gehostet und auf Microsoft Azure bereitgestellt, um Skalierbarkeit und Leistungsüberwachung zu gewährleisten.
- Azure bietet die notwendige Infrastruktur für das Hosting und Management des Empfehlungssystems.

Sie können dieses Projekt bei Bedarf gerne in Ihren internen Systemen verwenden.