# Projektplan: Entwicklung einer KI-Software zur Kundenbewertung mit PySpark

# Produktidee

Ein Marketingunternehmen möchte regelmäßig Kundendaten aus CSV-Dateien auswerten. Ziel ist es, mit Hilfe von PySpark MLlib vorherzusagen, ob ein Kunde ein Produkt kauft (gekauft = 1) oder nicht (gekauft = 0). Die Daten enthalten u. a.:  
- Alter  
- Einkommen  
- Geschlecht  
- BMI  
- Gekauft (Label)

Hier eine erste Idee für einen 2 Tage Plan:

**Sprint 1 (Tag 1):**

* CSV-Datenquellen laden
* Daten bereinigen und zusammenführen
* Explorative Analyse und Vorverarbeitung
* Feature Engineering

**🗓️ Sprint 2 (Tag 2):**

* Klassifikationsmodell trainieren
* Modell evaluieren
* Visualisierung der Ergebnisse
* Vorbereitung für Marketing-Ideen, welcher Personenkreis ist wichtig

Weiter noch die ML-Pipeline verbessern, ev. weitere Daten hinzunehmen.

Weiter noch das ML-Programm verbessern,

Lösungsideen:

ml\_projekt\_3\_spark.ipynb

(mit den Dateien : kunden.csv)

## Hinweise

- Optional können weitere Features wie BMI hinzugefügt oder das Modell mit RandomForestClassifier erweitert werden.  
- Erweiterungsidee: Hyperparameter-Tuning mit CrossValidation.

**Sprint 1 (Tag 1): Datenaufbereitung & Strukturierung**

* **Diskussion und Festlegung der Struktur der Quelldaten**  
  → Welche CSV-Dateien gibt es? Welche Features sind enthalten? Wie verknüpfen wir sie über Kunden-IDs?
* **Extraktion: CSV-Datenquellen laden und zusammenführen**  
  → Zwei Dateien mit unterschiedlichen Features anhand der Kunden-ID mergen.
* **Datenbereinigung und -vorverarbeitung**  
  → Fehlende Werte behandeln, kategorische Features (z. B. Geschlecht) numerisch umwandeln.
* **Feature Engineering**  
  → Relevante Features auswählen, ggf. neue Features (z. B. BMI) berechnen und hinzufügen.
* **Diskussion und Festlegung der Struktur des Codes**  
  → Modularisierung in Funktionen (ETL, Feature-Scaling, Modelltraining etc.)

**Sprint 2 (Tag 2): Modellentwicklung & Evaluation**

* **Auswahl und Implementierung verschiedener Modelle**  
  → Logistic Regression, Decision Tree (mit Gini-Impurity & Entropy), Random Forest
* **ML-Pipeline aufbauen mit Feature-Scaling und Cross-Validation**  
  → Pipeline erstellen, Modelle mit Parametergittern testen
* **Modelltraining und Evaluation**  
  → Modelle trainieren, Vorhersagen machen, Precision, Recall, F1-Score berechnen
* **Visualisierung der Ergebnisse**  
  → Balkendiagramme für Modellmetriken erstellen
* **Diskussion: Verbesserungsmöglichkeiten der Pipeline**  
  → Weitere Features hinzufügen, Hyperparameter-Tuning mit Cross-Validation
* **Vorbereitung für Marketing**  
  → Erkenntnisse über Zielgruppen ableiten und präsentieren