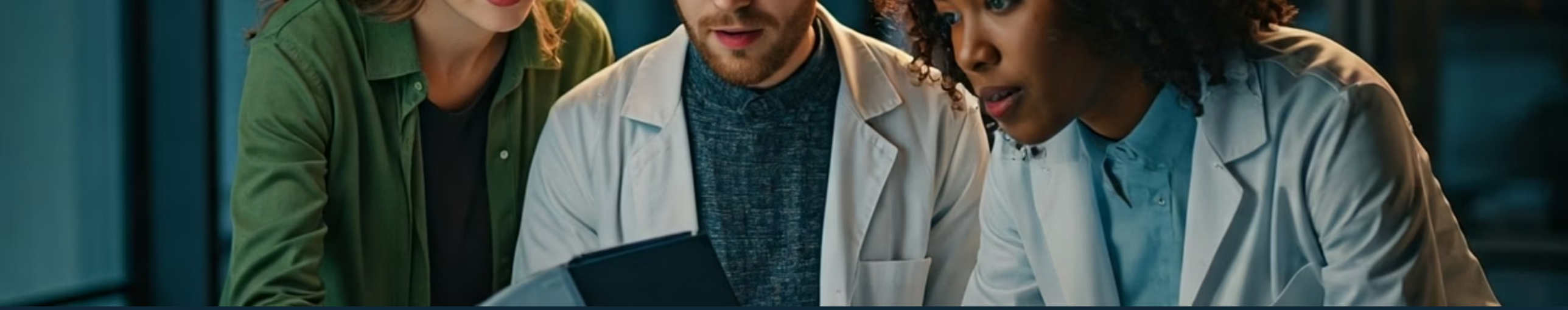
Präsentation: Kundenkaufvorhersage mit PySpark MLlib



Kundenkaufvorhersage: PySpark MLlib im Marketing

Ein agiles Data Science Projekt

Vorgestellt von:

- [Juri] (Projektübersicht)
- [Natalia] (Code-Struktur & implementierung)
- [Andrii] (Modell-Visualisierung & Ergebnisse)

Datum:

[15.06.2025]

Problemstellung & Lösung

Problem:

Marketing benötigt präzise Kundensegmentierung für Kampagnen.

Unsere Lösung:

KI-Modell (PySpark MLlib) zur Vorhersage des Kaufverhaltens (gekauft = 1 vs. 0).

Ihr Nutzen:

Effizientere Kampagnen, höhere Konversionsraten.

Datenbasis & Ansatz

1

Datenquellen:

kunden.csv, bmi.csv (Features: Alter, Einkommen, Geschlecht, BMI. Label: gekauft).

2

Vorgehen:

Agiles Projektmanagement nach Scrum (2 Sprints).

3

Sprint 1:

Datenaufbereitung & Feature Engineering.

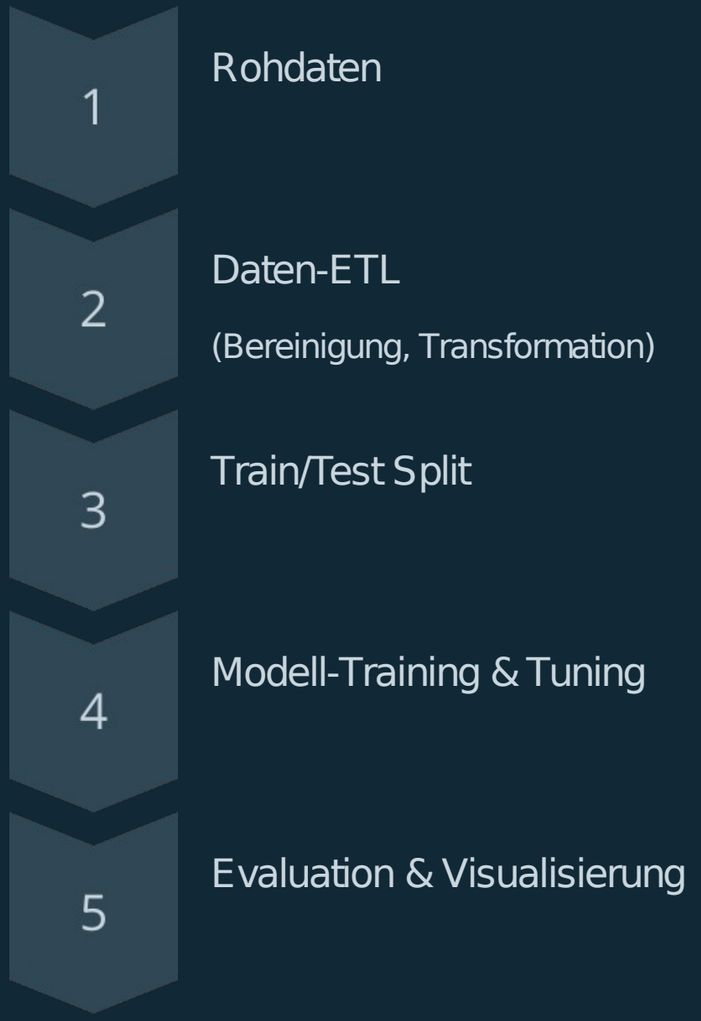
4

Sprint 2:

Modellentwicklung & Evaluation.

End-to-End ML-Pipeline: Der Workflow

Visuell: Klares, schematisches Blockdiagramm des Prozesses:



Datenvorbereitung & ETL-Prozess

Daten-ETL in PySpark: Rohdaten → ML-bereite Features

extract_and_merge_data():

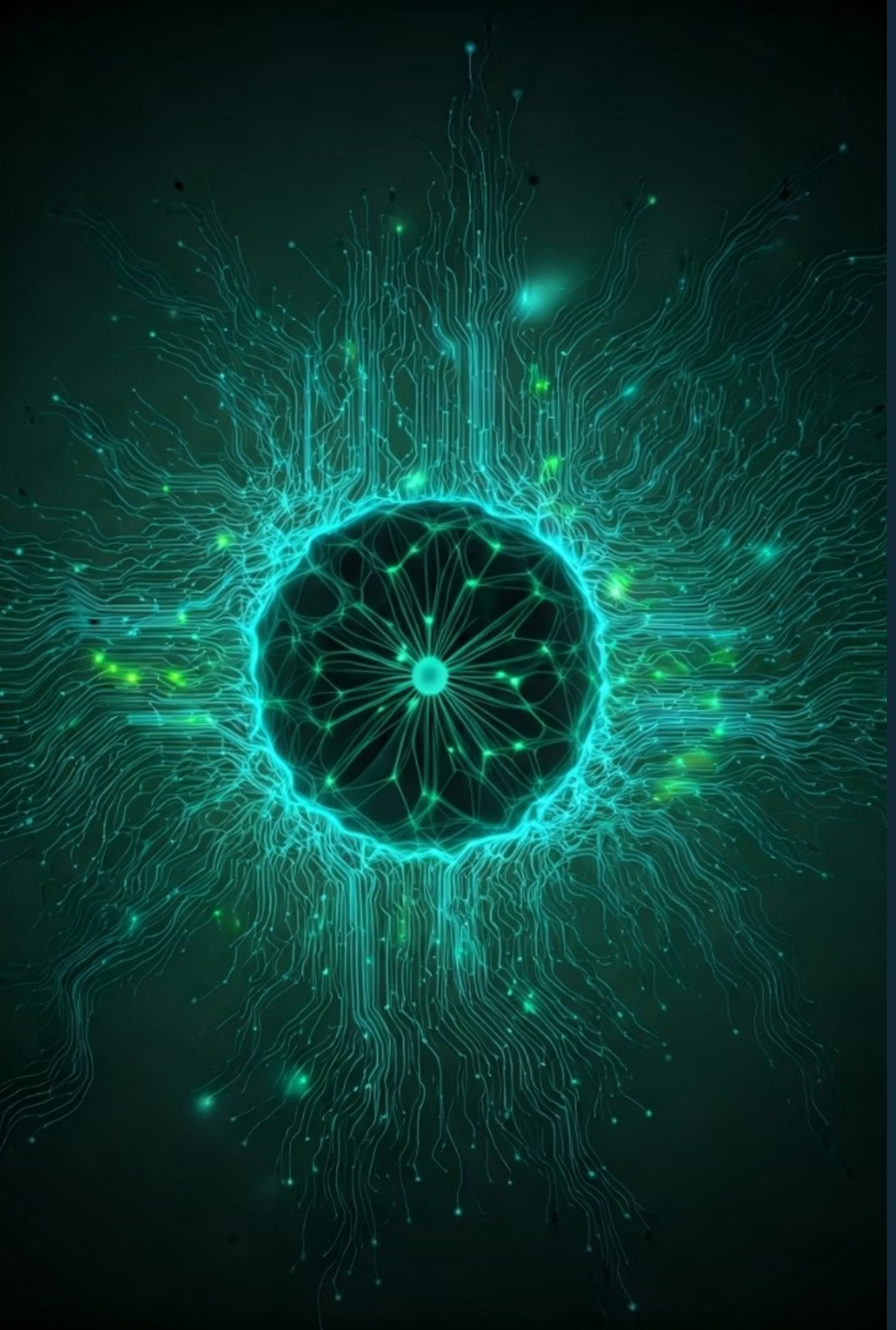
Laden & Verknüpfen von Kunden- & BMI-Daten.

transform_data():

Fehlende Werte behandeln, kategoriale Features umwandeln, Feature-Vektor erstellen.

scale_features():

Numerische Features skalieren.



Modell-Definition & Trainings-Setup

Modellierung in PySpark MLlib:

prepare_train_test_split():

Datenaufteilung (Train/Test).

build_models():

Definition von Logistic Regression, Decision Tree, Random Forest. (Inkl. Hyperparameter-Gitter für Cross-Validation.)

main()-Funktion:

Orchestrierung der gesamten PySpark-Pipeline.

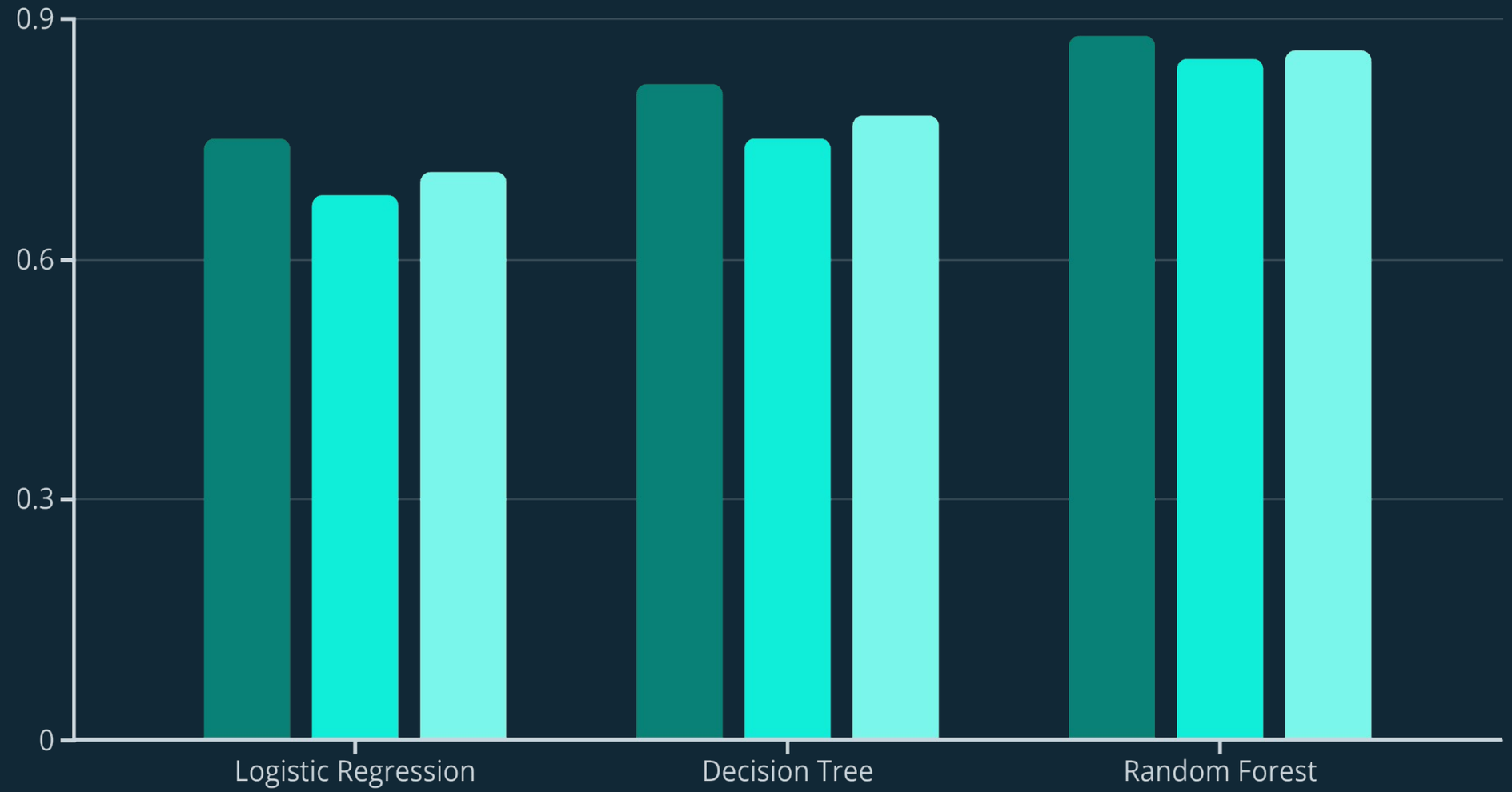
Modell-Evaluation & Ergebnisse

Bewertung: Precision, Recall, F1-Score.

Methode: Modell-Evaluation mit PySpark MLlib.

Fazit: RandomForestClassifier ist das beste Modell.

Visuell: Screenshot Ihres Barplots model_metrics.png (des Vergleichs aller Modelle).



Detaillierte Analyse des besten Modells

Modell: RandomForestClassifier

0.75	1.00	0.78
Precision	Recall	F1-Score

Einblicke: ROC-Kurve, Konfusionsmatrix, Feature Importance (Hinweis: werden im Code gezeigt).

Schlussfolgerung, Business Value & Ausblick

Fazit: Effiziente, skalierbare PySpark MLlib Pipeline.

Nutzen: Datenbasiertes, zielgerichtetes Marketing.

Unser Beitrag: Umfassende Realisierung & Vergleich in PySpark.

"Als Nächste Schritte schlagen wir vor: das Deployment der Lösung, beispielsweise als **API-Service**; ein kontinuierliches Retraining mit neuen Daten; und A/B-Tests, um den **realen Einfluss in der Praxis zu messen**."