

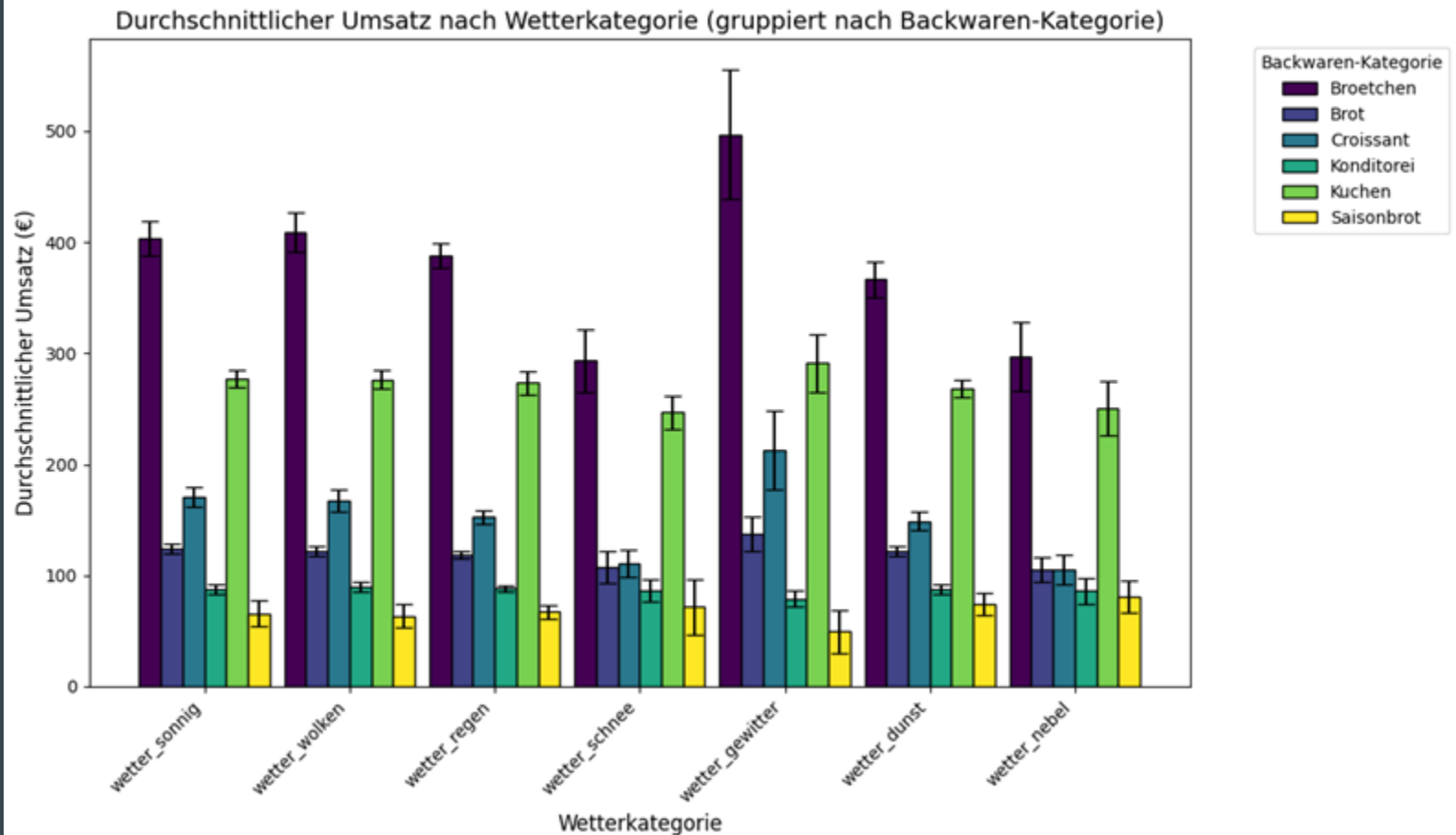
Einführung in Data Science und maschinelles Lernen

...

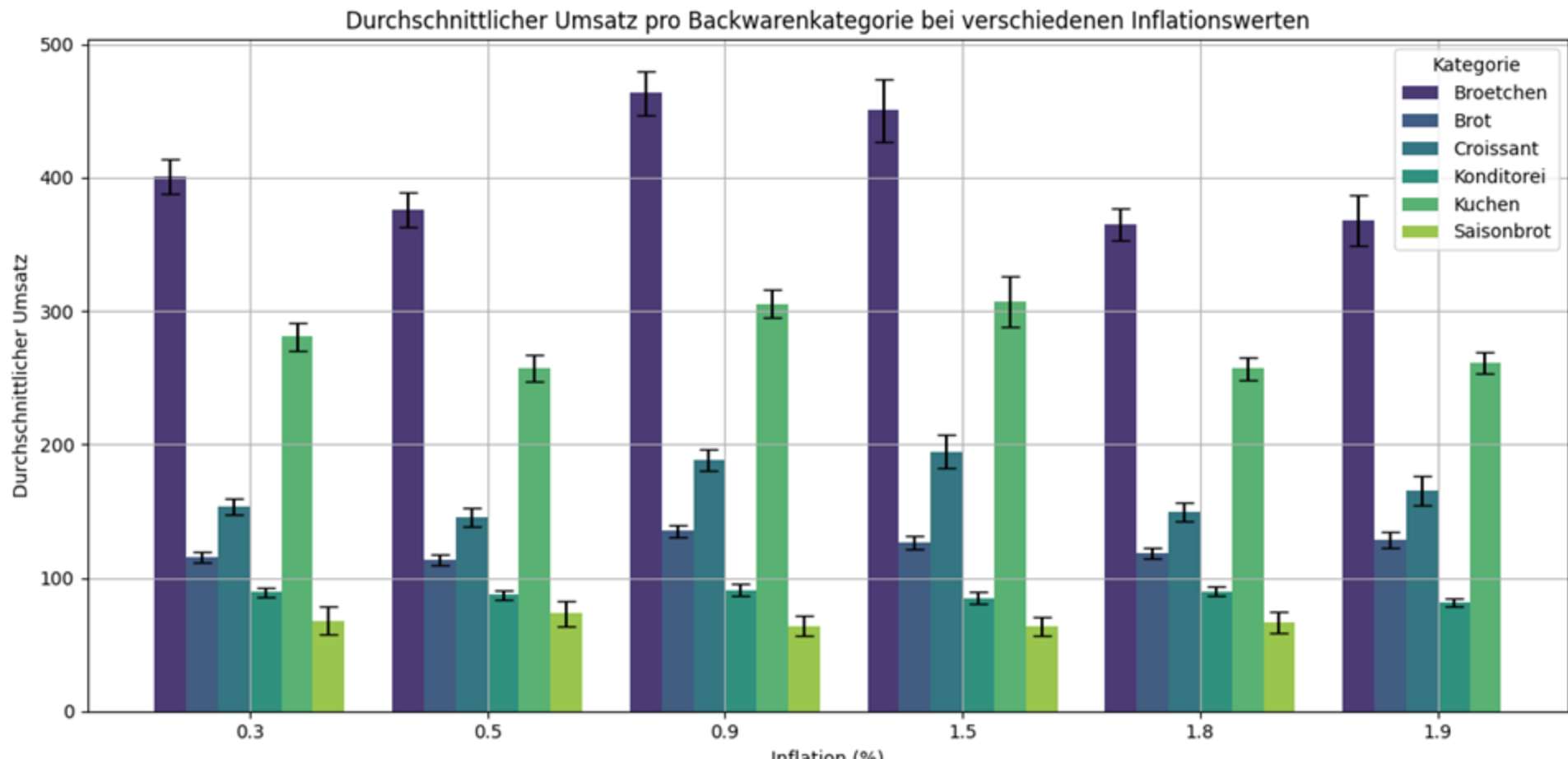
Gruppe 6: Isabel Kremin, Heidrun Schwalowski & Moritz Hintringer

Datum: 09.01.2024

Charakteristiken des Datensatzes



Charakteristiken des Datensatzes



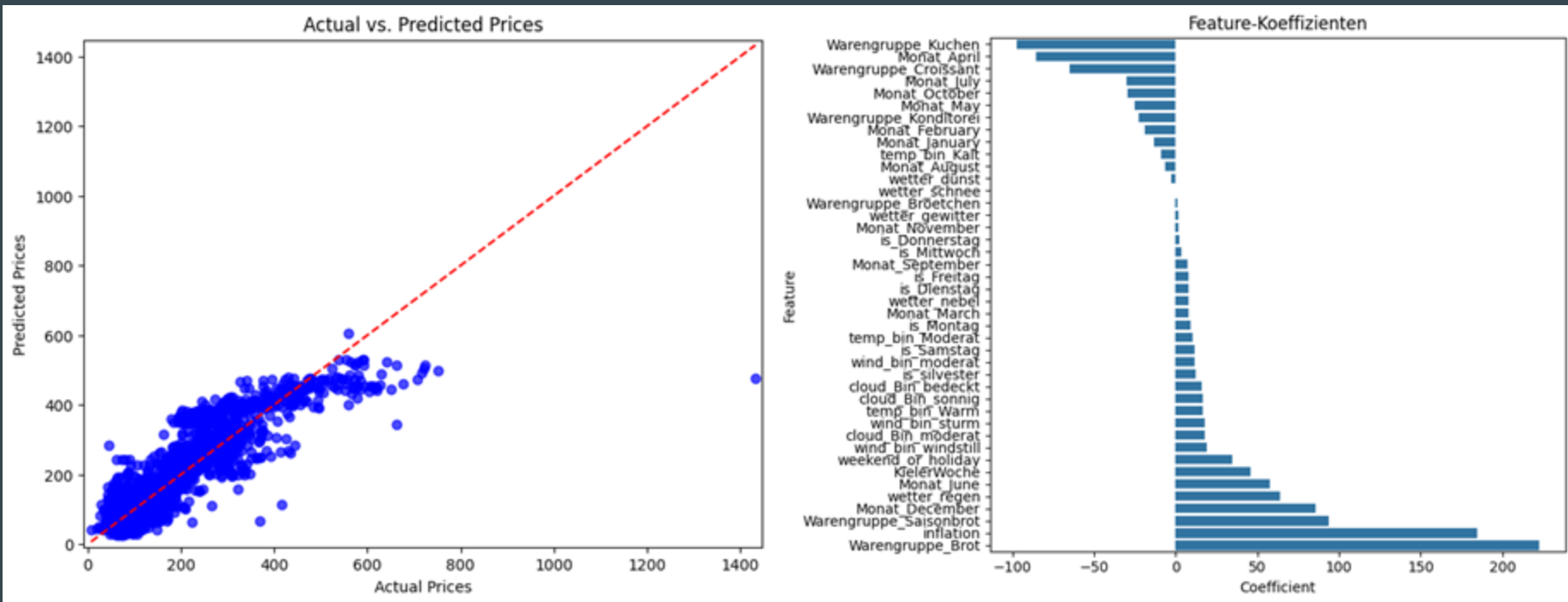
Optimierung des linearen Modells - SGD Regression

Modellgleichung:

```
sgd_model = SGDRegressor(max_iter=1000, learning_rate='invscaling',  
eta0=0.01, random_state=42, alpha=0.001, penalty='l2')
```

- 42 Feature
- Adjusted r^2 : 0.7238
- Mean absolute percentage error: 32.39%

Optimierung des linearen Modells - SGD Regression



Modelldefinition und Evaluierung

```
# create model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(128,
                           input_shape=(len(features),),
                           activation='relu',
                           kernel_regularizer=l2(0.001)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(0.3),

    tf.keras.layers.Dense(128,
                           activation='relu',
                           kernel_regularizer=l2(0.001)),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(0.3),

    tf.keras.layers.Dense(32,
                           activation='relu',
                           kernel_regularizer=l2(0.001)),
    tf.keras.layers.Dense(1)
])

# Definiere Huber-Loss with specific delta value
loss = Huber(delta=25)
```

```
# learning rate scheduler
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(
    initial_learning_rate=0.0005,
    decay_steps=5000,
    decay_rate=0.9)

# model compilation
model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=lr_schedule),
    loss=loss,
    metrics=['mae']
)

# early stopping
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=35,
    restore_best_weights=True
)

# training
history = model.fit(
    x_train, y_train,
    validation_data=(x_val, y_val),
    epochs=150,
    batch_size=32,
    callbacks=[early_stopping],
    verbose=1
)
```

Modelldefinition und Evaluierung



Modelldefinition und Evaluierung: Trainingsdaten

MAPE on the Training Data: 16.26%

MAPE für Brot: 18.06%

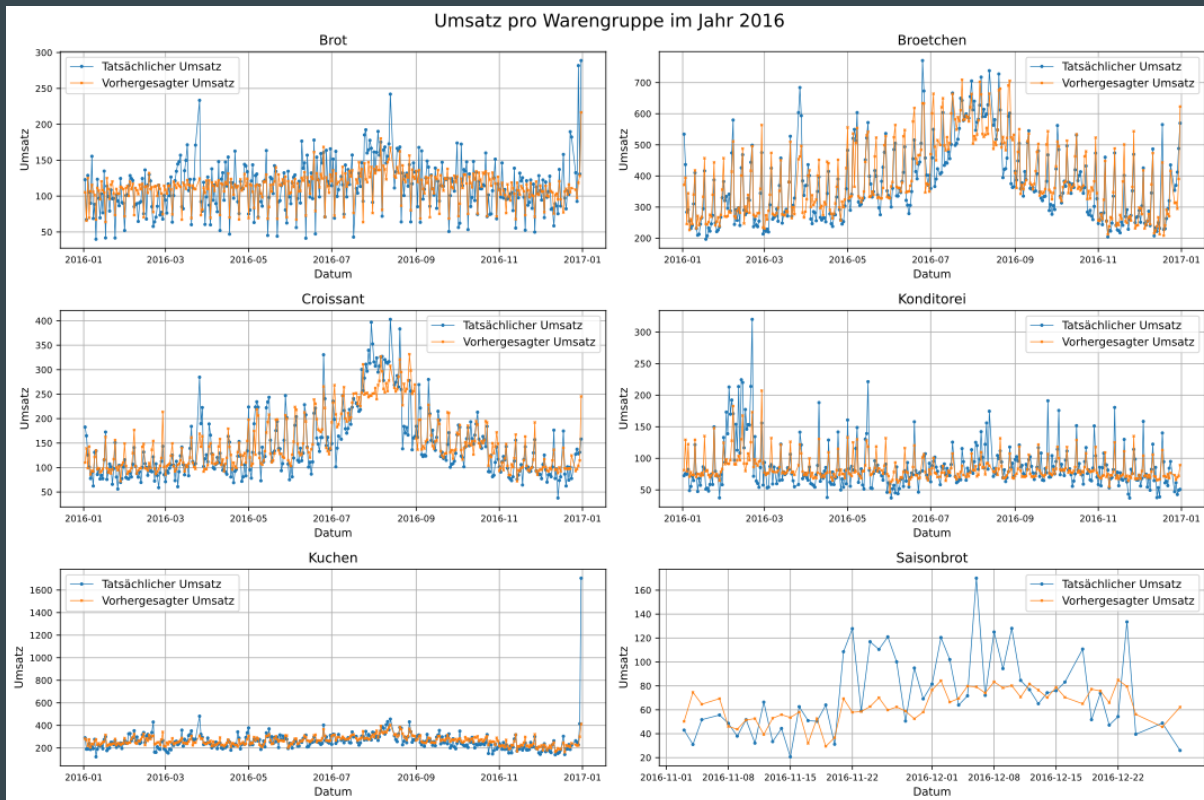
MAPE für Broetchen: 10.87%

MAPE für Croissant: 16.15%

MAPE für Konditorei: 20.43%

MAPE für Kuchen: 11.89%

MAPE für Saisonbrot: 41.35%



Ergebnisse: Validierungsdaten

MAPE on the Training Data: 16.26%

MAPE für Brot: 18.06%

MAPE für Broetchen: 10.87%

MAPE für Croissant: 16.15%

MAPE für Konditorei: 20.43%

MAPE für Kuchen: 11.89%

MAPE für Saisonbrot: 41.35%

MAPE on the Validation Data: 19.22%

MAPE für Brot: 19.55%

MAPE für Broetchen: 13.53%

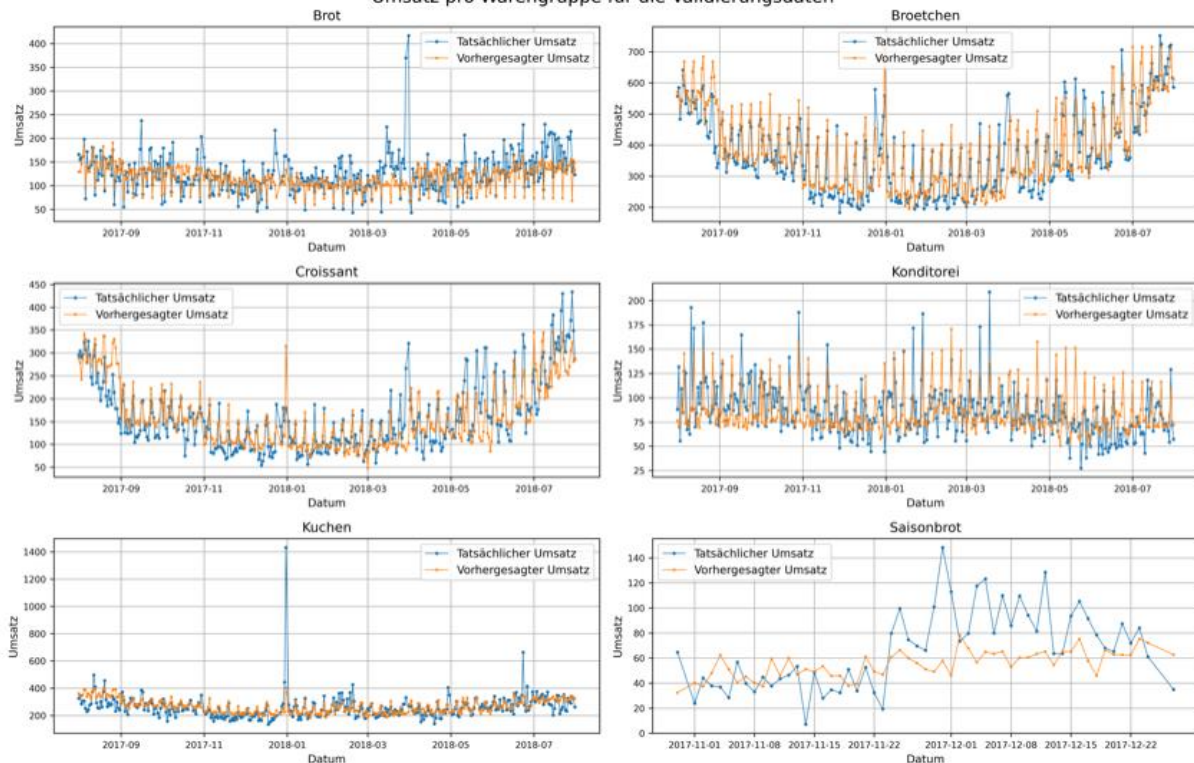
MAPE für Croissant: 20.67%

MAPE für Konditorei: 22.66%

MAPE für Kuchen: 15.68%

MAPE für Saisonbrot: 44.82%

Umsatz pro Warengruppe für die Validierungsdaten



Challenges und Fehler

- Vermeidung von Overfitting
- Welches Modell ist am Besten?
- Jupyter Kernel Unterbrechungen im Codespace
- Feature Engineering

Danke für eure Aufmerksamkeit!