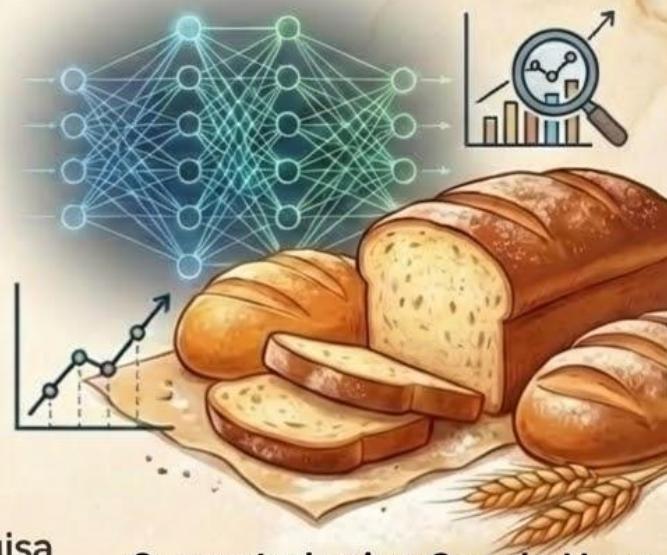


# Bakery Sales Prediction Project: From Linear Regression to Neural Networks

Course name: Introduction to Data Science & Machine Learning



Group number 3:  
Akshay Patil, Andrews Mensah, Louisa  
Lagmoeller, Aadip Thapaliya

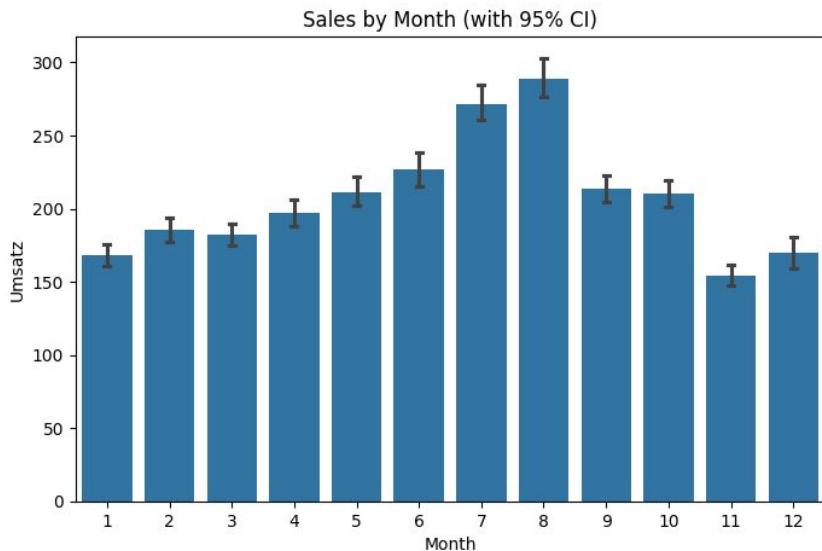
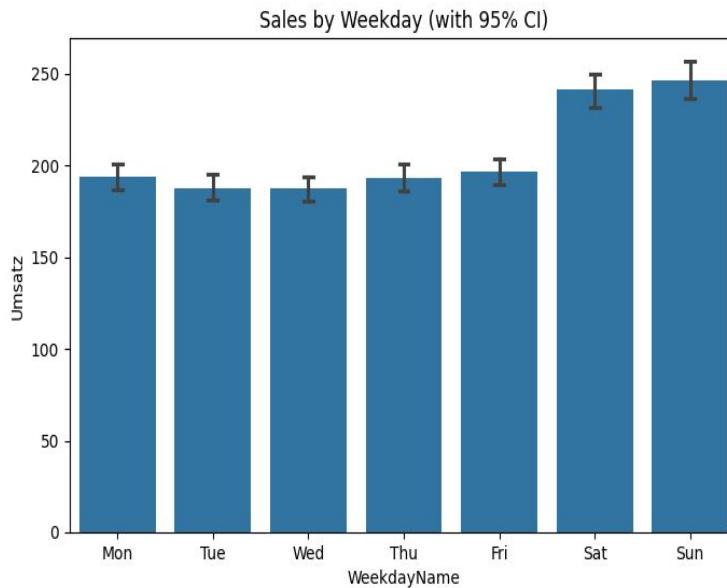


Generated using Google Nano  
Banana Model (6th Jan, 2026)

## List and brief description of self-created variables

- **Overview:** To improve our model's ability to capture seasonality and trends, we engineered several features from the raw **Datum** column.
- **Variables:**
  - **Year, Month, Day:** Extracted to capture annual growth and monthly seasonality.
  - **Weekday:** Integer values were assigned (0=Monday, 6=Sunday) to capture weekly cycles.
  - **IsWeekend:** Boolean flag (True for Saturday/Sunday) to capture the high traffic on weekends.
  - **KielerWoche:** Binary flag (1 during the event, 0 otherwise). Accounts for the massive influx of visitors during the festival.

# Bar charts with confidence intervals for two self-created variables



# Baseline Model - Linear Regression

Model\_Equation:

```
Sales = β₀ - 70.68 (Warengruppe_1) + 218.58 (Warengruppe_2) - 27.87 (Warengruppe_3) - 103.27  
(Warengruppe_4) + 88.88 (Warengruppe_5) - 105.64 (Warengruppe_6) + 47.88 (IsWeekend) + 1.13 (Weekday) +  
14.84 (KielerWoche) + 3.92 (Temperature) + 0.15 (WindSpeed) - 0.08 (CloudCover) + 0.01 (WeatherCode) -  
11.48 (Year) - 1.06 (Month) + 0.35 (Day)
```

## Key Drivers:

- Product Type: The strongest predictor (Group 2 sells +218€ vs baseline).
- Seasonality: Weekends (+48€) and Events (+15€) are significant positive drivers.

## Model Performance:

- Validation Adjusted R²: 0.6859 (Strong Baseline).
- Validation MAPE: 32.12%.
- Conclusion: The model explains ~69% of the variance in sales, proving that basic features (Product + Time) work well. However, a 32% average error is still too high for precise inventory planning.

## Type of missing value imputation used

**Challenge:** The **weather** dataset and **Kieler Woche** contained missing entries.

**Imputation Strategy:**

1. **KielerWoche & Wettercode Dataset:** Filled **NaN** with **0** (Assumed "No Event" / "No significant weather" if missing). **Reasoning:** This is a structural decision, not a guess. Since the original files only listed dates when the festival or a weather event *actually occurred*, any date that didn't match during our merge is logically a 'non-event' day. Therefore, treating **NaN** as **0** correctly represents 'No Festival' or 'Normal Weather'.
2. **Weather Imputation (Temp, Wind, Clouds):** Used **Forward Fill** followed by **Backward Fill**. **Reasoning:** Weather conditions (like temperature or cloud cover) typically persist over short periods. Using the last known value is a safe, logical baseline (Persistence Model).

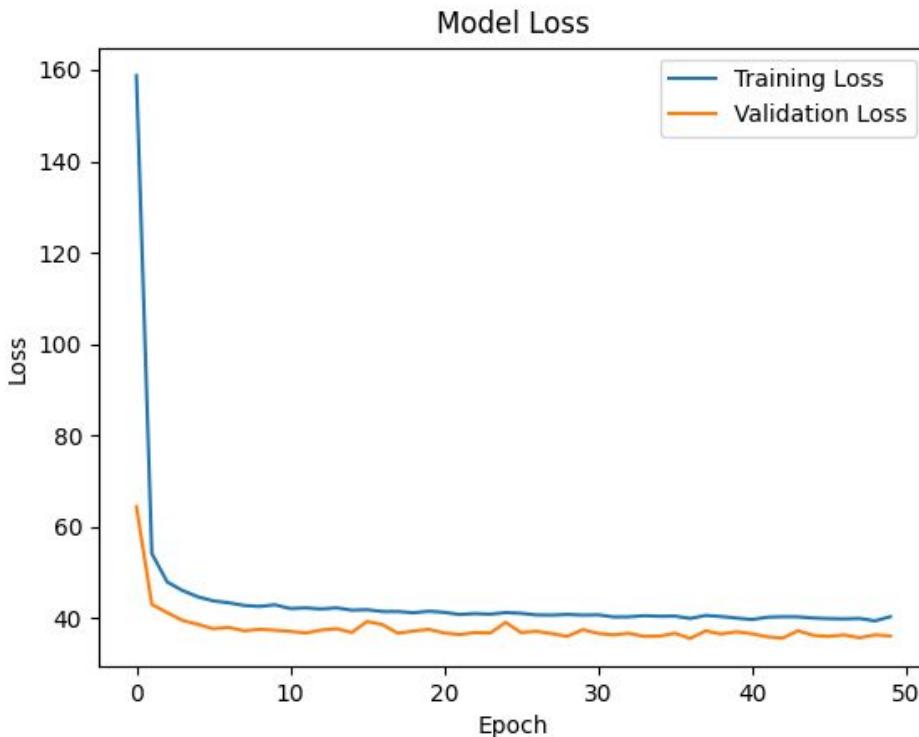
# Neural network model

```
# Define Model
model = keras.Sequential([
    layers.Dense(64, activation='relu', input_shape=[X_train.shape[1]]), # Input
    Layer matching feature dimensions
    layers.Dropout(0.2), # Dropout layer to prevent Overfitting
    layers.Dense(32, activation='relu'), # Hidden Layer
    layers.Dense(1) # Output Layer ( Single value for Regression)
])

model.compile(
    optimizer='adam',
    loss='mae', # Mean Absolute Error is often good for regression
)

# Train
history = model.fit(
    X_train_scaled, y_train,
    validation_data=(X_val_scaled, y_val),
    batch_size=32,
    epochs=50,
    verbose=1
)
```

# Neural Network Training



## Observations:

- The training loss decreases steadily, indicating the model is learning.
- The validation loss stabilizes around 50 epochs, suggesting the model has converged without significant overfitting (thanks to the Dropout layer).

## **Neural Network Results (MAPE)**

### **Overall Validation Performance:**

- **Neural Net Validation MAPE: 0.2018 (20.18% ).** On average, our predictions are off by ~20.18%)

### **Performance by Product Group:**

- **Warengruppe 1 (Bread): 0.2342 (23.4%)**
- **Warengruppe 2 (Rolls): 0.1380 (13.8%) (Best performing group)**
- **Warengruppe 3 (Croissant): 0.2042 (20.4%)**
- **Warengruppe 4 (Confectionery): 0.2394 (23.9%)**
- **Warengruppe 5 (Cake): 0.1451 (14.5%)**
- **Warengruppe 6 (Seasonal Bread): 0.5812 (58.1%) (Hardest to predict)**

## "Worst Fail" vs. "Best Improvement"

### ❖ **Worst Fail (The Model's Blind Spot)**

- Date: 2017-12-31 (New Year's Eve)
- Actual Sales: 1,432.42 €
- Predicted: 262.21 €
- Error: 1,170.21 € (Underestimated by ~82%)
- Diagnosis: The model treated this as a regular Sunday. It failed to capture the massive, irregular demand spike specific to New Year's Eve.

### ❖ **Best Improvement (Model Strength)**

- Date: 2017-08-13 (Summer Sunday)
- Actual Sales: 537.19 €
- Predicted: 515.50 €
- Error: 21.69 € (Only ~4% Error)
- Success Factor: Unlike the linear baseline, the Neural Network correctly identified the non-linear interaction between "Sunday" and "Summer Weather," predicting the high volume almost perfectly.

# Conclusion

## Summary:

- Transitioning from Linear Regression to Neural Networks improved prediction accuracy by **~37%** (MAPE dropped from **32.1%** to **20.2%**).

## Key Learnings:

- Feature Engineering: Simple flags (Weekends/Holidays) were more impactful than complex raw data.
- Non-Linear Patterns: Neural Networks successfully captured complex interactions (e.g., Summer + Weekend) that linear models missed.

## Next Steps:

- Incorporate "**School Holidays**" and specific "**Public Holidays**" (like New Year's Eve) as explicit features to fix the "Worst Fail" cases.