Final Presentation
Group 4

03.07.2025

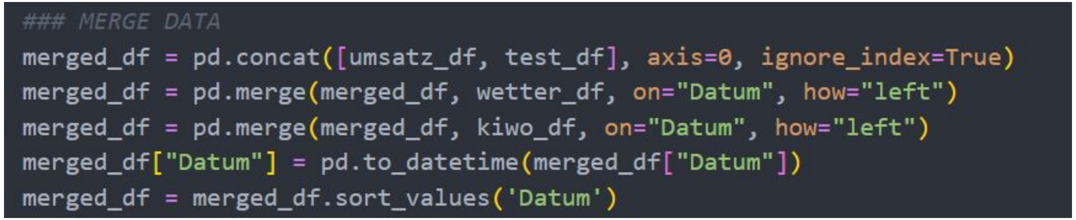
Group members:

- Hauke Beyer
- Anil Kumar Gadamoni

Data Characteristics

Data Preparation

- Importing Libraries and loading the data from the server
- Merging the data

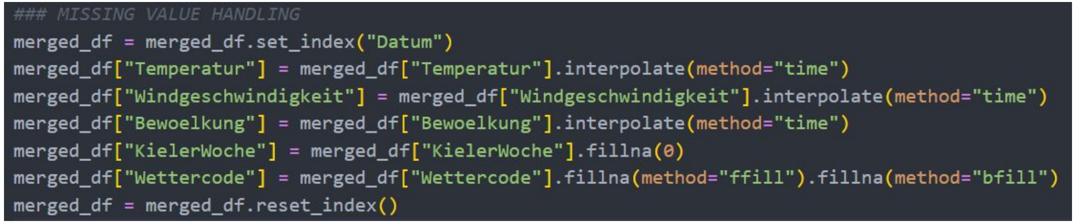


```
### MERGE DATA
merged_df = pd.concat([umsatz_df, test_df], axis=0, ignore_index=True)
merged_df = pd.merge(merged_df, wetter_df, on="Datum", how="left")
merged_df = pd.merge(merged_df, kiwo_df, on="Datum", how="left")
merged_df["Datum"] = pd.to_datetime(merged_df["Datum"])
merged_df = merged_df.sort_values('Datum')
```

Data Characteristics

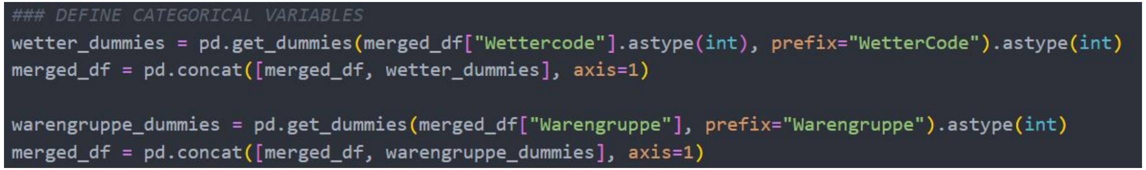
Data Preparation

- Data Cleaning and Imputing



```
### MISSING VALUE HANDLING
merged_df = merged_df.set_index("Datum")
merged_df["Temperatur"] = merged_df["Temperatur"].interpolate(method="time")
merged_df["Windgeschwindigkeit"] = merged_df["Windgeschwindigkeit"].interpolate(method="time")
merged_df["Bewoelkung"] = merged_df["Bewoelkung"].interpolate(method="time")
merged_df["KielerWoche"] = merged_df["KielerWoche"].fillna(0)
merged_df["Wettercode"] = merged_df["Wettercode"].fillna(method="ffill").fillna(method="bfill")
merged_df = merged_df.reset_index()
```

- Defining categorical variables



```
### DEFINE CATEGORICAL VARIABLES
wetter_dummies = pd.get_dummies(merged_df["Wettercode"].astype(int), prefix="WetterCode").astype(int)
merged_df = pd.concat([merged_df, wetter_dummies], axis=1)

warengruppe_dummies = pd.get_dummies(merged_df["Warengruppe"], prefix="Warengruppe").astype(int)
merged_df = pd.concat([merged_df, warengruppe_dummies], axis=1)
```

- Splitting and saving

Data Characteristics

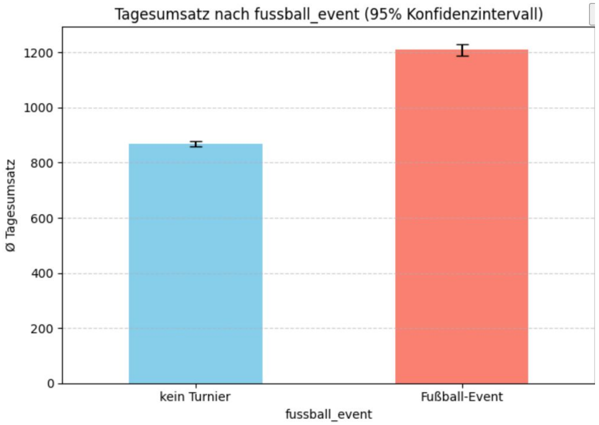
Additional Variables

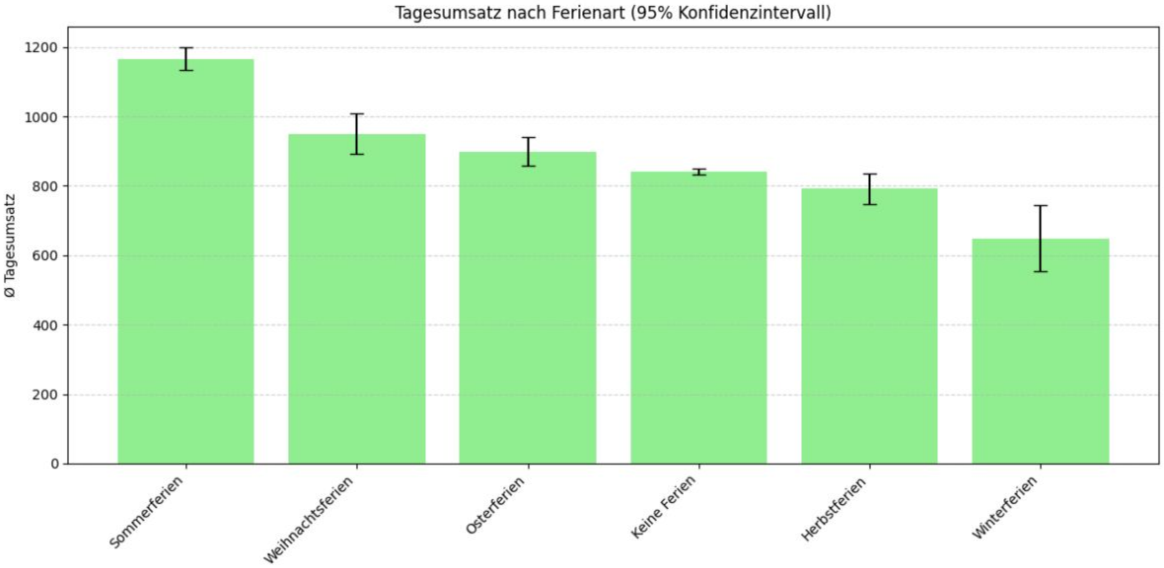
- National/Regional Holidays and the days before (Christmas Eve, etc) (`import holidays`)
- DAX (`import yfinance as yf1`)
- Weekday
- Sunhours (Sunrise - Sunset)... (`from astral import sun from astral import Observer obs = Observer(latitude=54.3233,longitude=10.1228)`)
- Major Football Events (`wm_2014 = pd.Timestamp('2014-06-12') <= datum <= pd.Timestamp('2014-07-13')`)
- School Vacation (`{"name": "Sommerferien", "start": "2013-06-24", "end": "2013-08-03"},`)

Feature Engineering

- Average Sales for a typical day of the week x in month y for the different product groups
- Average Sales on day n of the year averaged over the whole Iota

Bar charts for two self-created variables





Baseline Model

- Built a **Linear Regression model** to predict sales (Umsatz) using both **categorical** (e.g., Warengruppe, Feiertag) and **numerical** features (e.g., Temperatur, Bewoelkung).

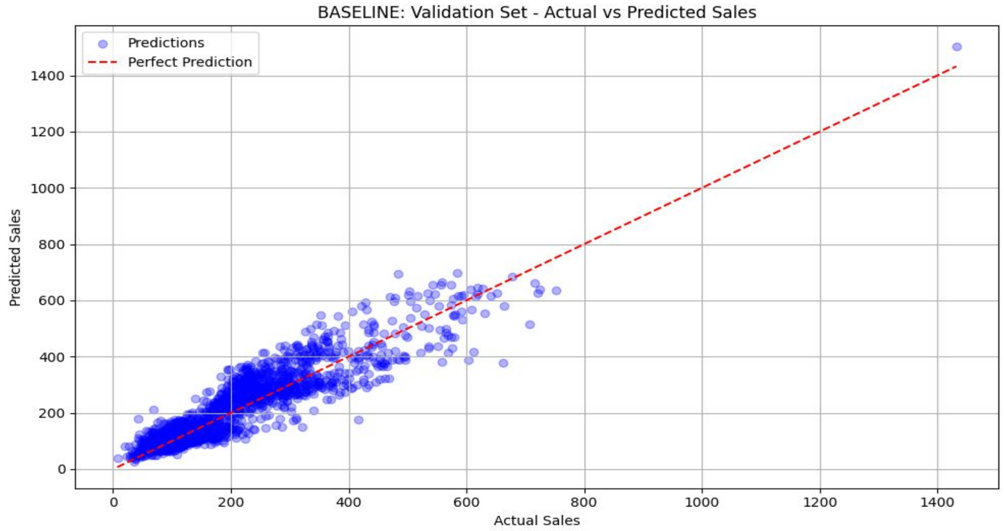
- Applied a **preprocessing pipeline**: imputed missing values, scaled numerical data, and one-hot encoded categorical variables.

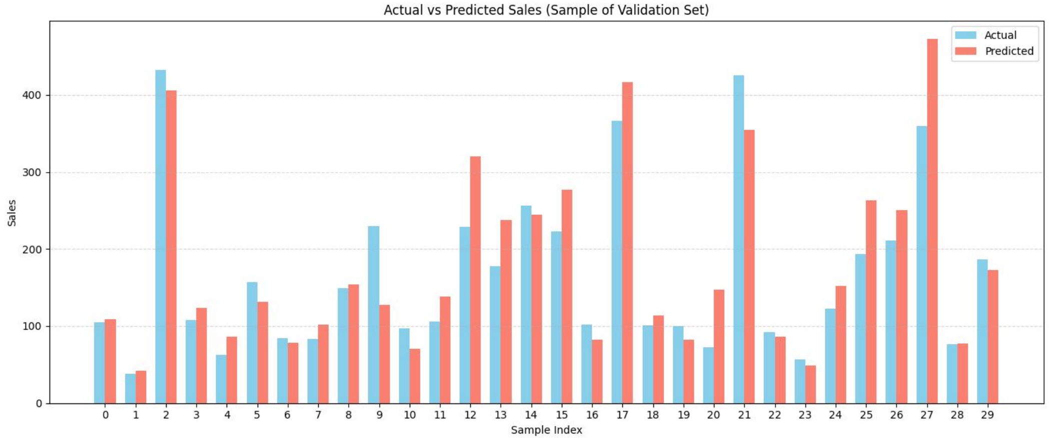
- Combined preprocessing and modeling in a **scikit-learn pipeline** for clean, consistent, and reproducible training and prediction.

Baseline Model

- The **scatter plot** shows a strong linear relationship between actual and predicted sales, with most points close to the ideal prediction line — indicating good overall model accuracy.

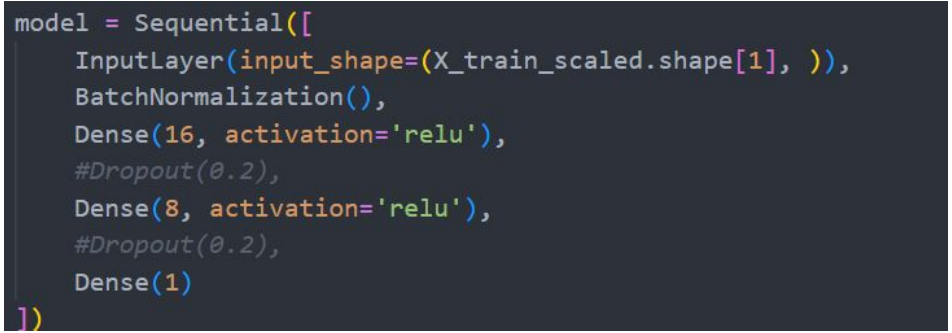
- The **bar chart** confirms this by comparing actual and predicted values for a sample of cases, where most bars are closely aligned.





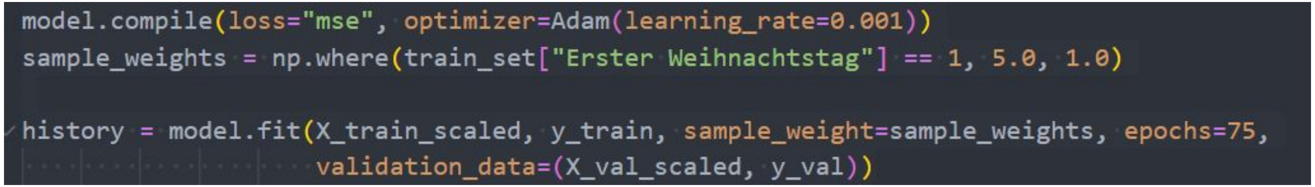
Neural network Definition

Source code defining the neural network:



```
model = Sequential([
    InputLayer(input_shape=(X_train_scaled.shape[1], )),
    BatchNormalization(),
    Dense(16, activation='relu'),
    #Dropout(0.2),
    Dense(8, activation='relu'),
    #Dropout(0.2),
    Dense(1)
])
```

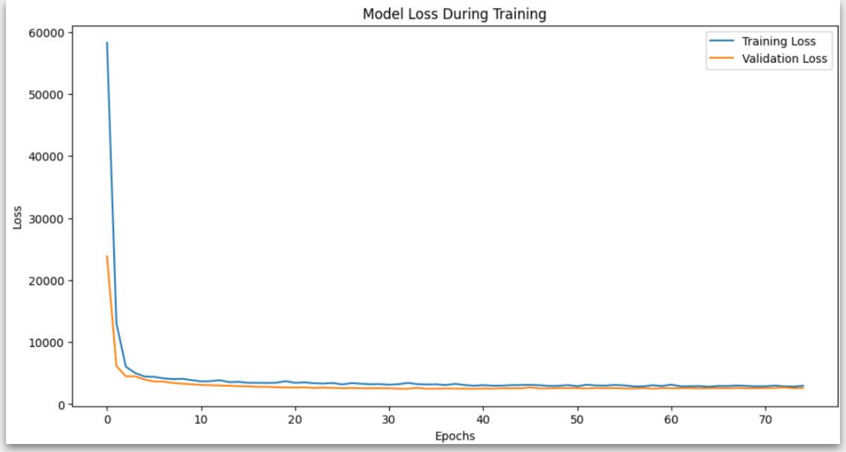
Training



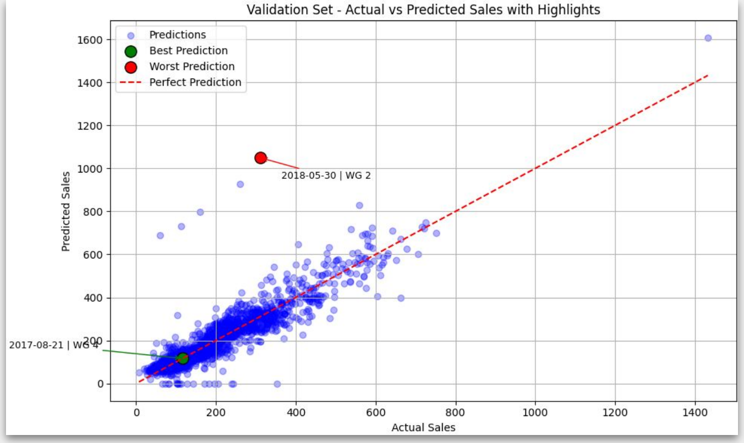
```
model.compile(loss="mse", optimizer=Adam(learning_rate=0.001))
sample_weights = np.where(train_set["Erster Weihnachtstag"] == 1, 5.0, 1.0)

history = model.fit(X_train_scaled, y_train, sample_weight=sample_weights, epochs=75,
                    validation_data=(X_val_scaled, y_val))
```

Neural network Evaluation 😊

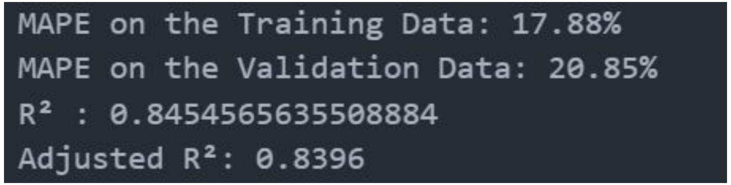
Loss function



Best and worst predictions



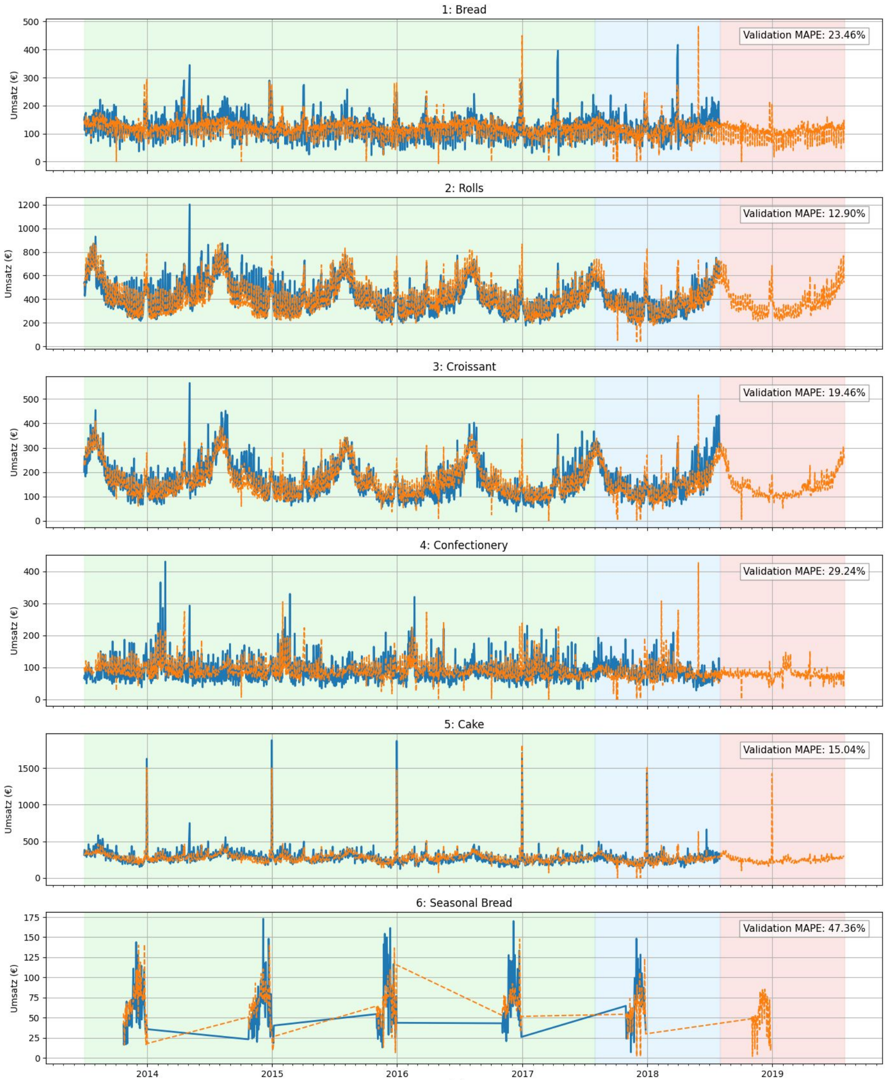
MAPE:



```
MAPE on the Training Data: 17.88%
MAPE on the Validation Data: 20.85%
R² : 0.845456563508884
Adjusted R²: 0.8396
```

Neural network

- Good prediction of the validation data
- Still lacks ‘variance’ and shows some weird dips



Challenges and Errors

- Bugs in the code
- Adding more variance in the model
- Defining reasonable additional variables
- Improving MAPE value

Q & A