Bakery Sales Prediction

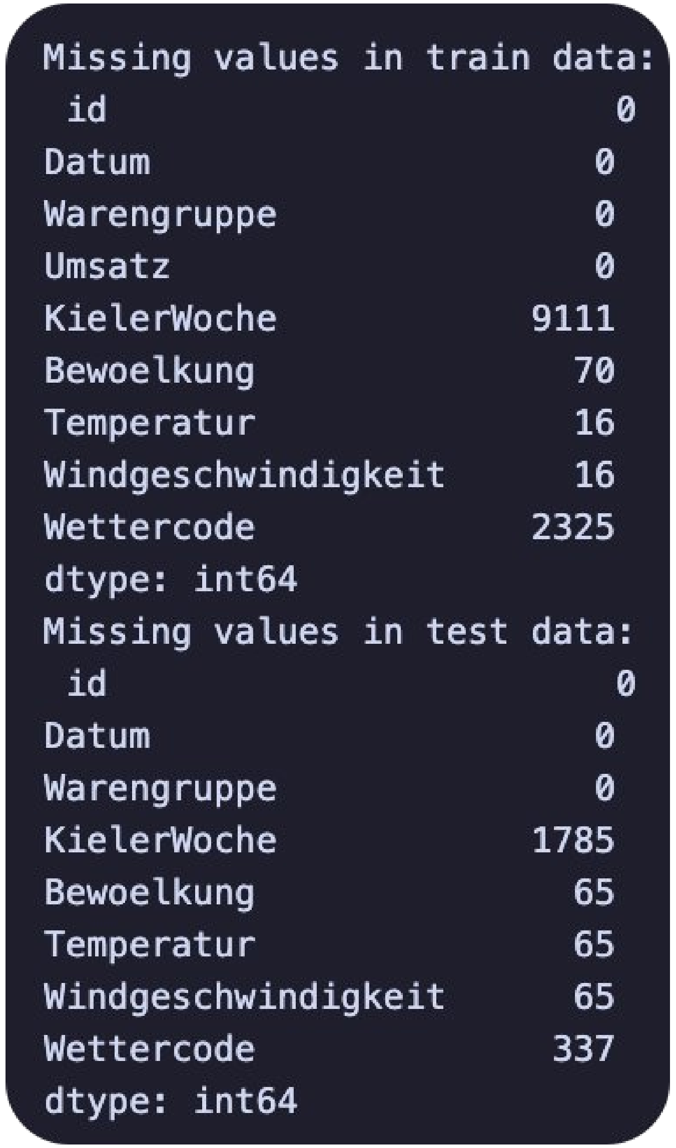
Group 2 presentation: Introduction to Datascience and Machine Learning

Lukas Kling, Lina Sandberg, Edil, Melissa Muszelewski

Data imputation

Missing Values

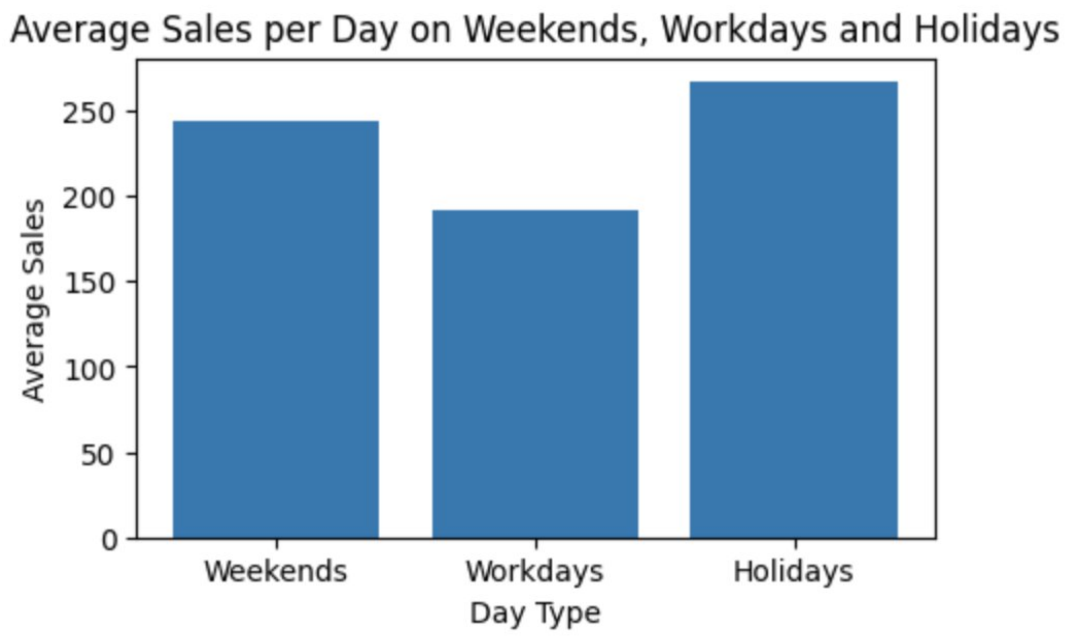
- KiWo
- Weather Data
    - Temperatur
    - Windgeschwindigkeit
    - Bewölkung
    - Wettercode
- Methods
    - Binary categories
    - Average
    - Copy from Day/Week/Month before

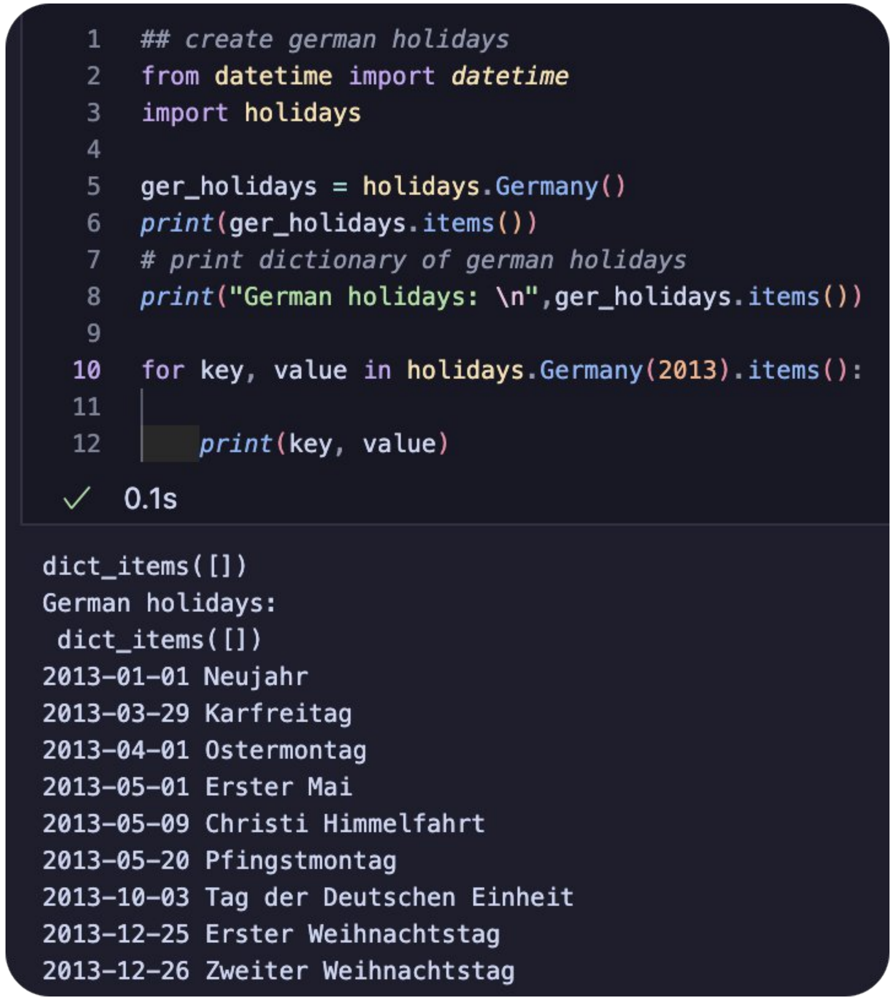


```
Missing values in train data:
 id                      0
Datum                    0
Warengruppe              0
Umsatz                   0
KielerWoche           9111
Bewoelkung              70
Temperatur              16
Windgeschwindigkeit     16
Wettercode            2325
dtype: int64
Missing values in test data:
 id                      0
Datum                    0
Warengruppe              0
KielerWoche           1785
Bewoelkung              65
Temperatur              65
Windgeschwindigkeit     65
Wettercode             337
dtype: int64
```

Data exploration

- Impact of
  - Weekdays
  - Weekends
  - Public Holidays





```python
## create german holidays
from datetime import datetime
import holidays

ger_holidays = holidays.Germany()
print(ger_holidays.items())
# print dictionary of german holidays
print("German holidays: \n",ger_holidays.items())

for key, value in holidays.Germany(2013).items():

    print(key, value)
```

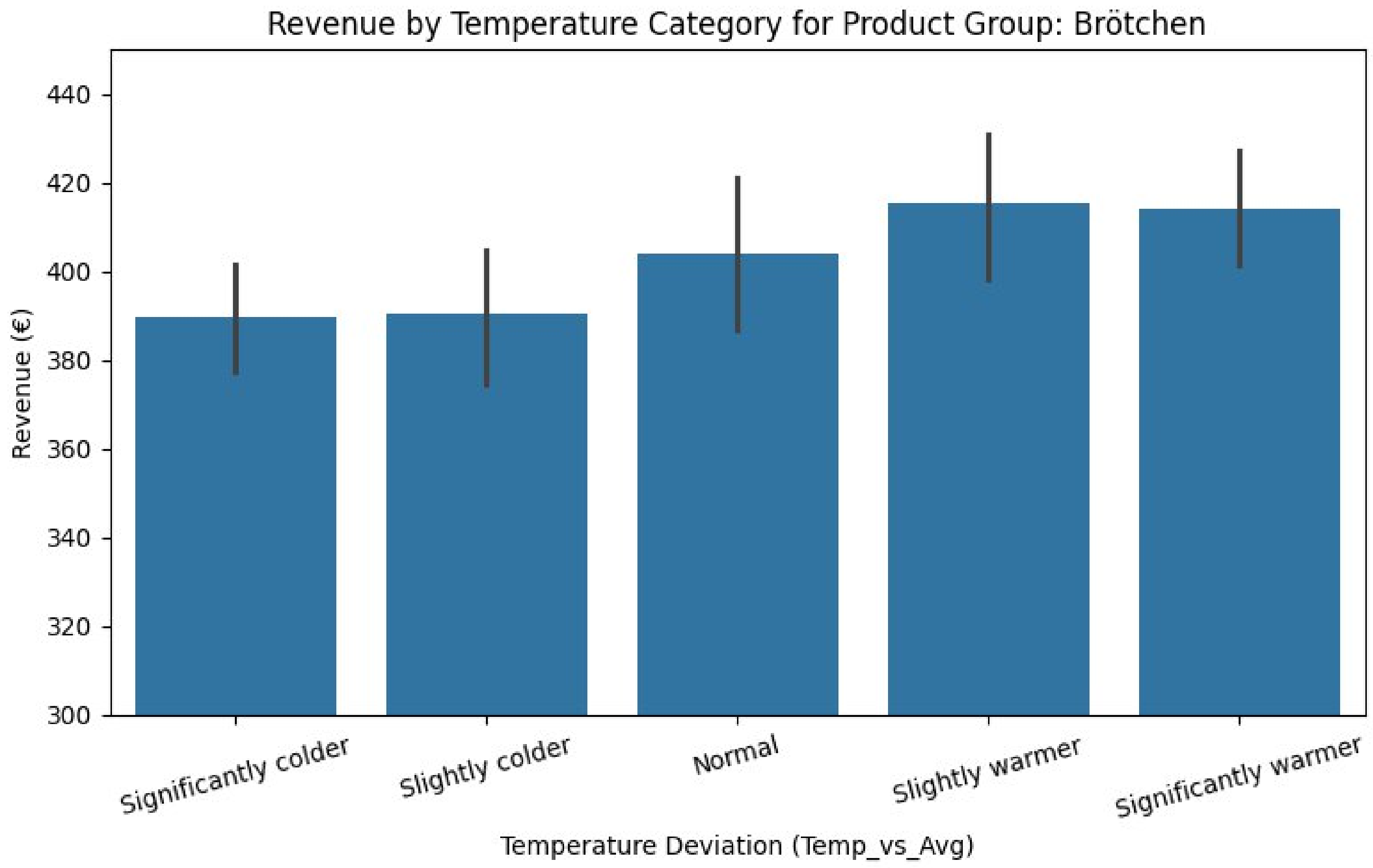
✓ 0.1s

```
dict_items([])
German holidays:
 dict_items([])
2013-01-01 Neujahr
2013-03-29 Karfreitag
2013-04-01 Ostermontag
2013-05-01 Erster Mai
2013-05-09 Christi Himmelfahrt
2013-05-20 Pfingstmontag
2013-10-03 Tag der Deutschen Einheit
2013-12-25 Erster Weihnachtstag
2013-12-26 Zweiter Weihnachtstag
```

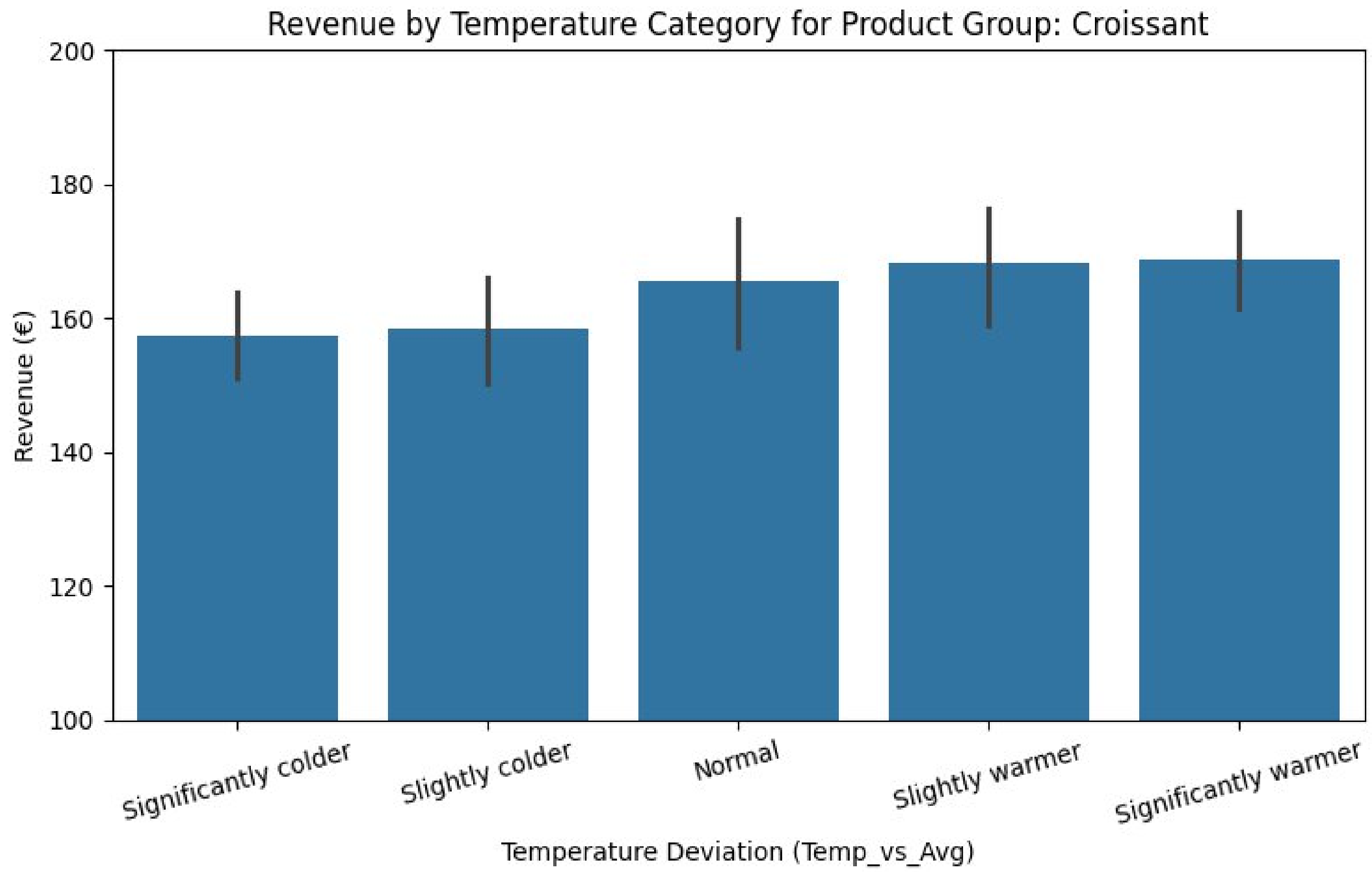
Self created variables

- Weekday
- Is_holiday
  - added all the holidays
- Temp_vs_Avg
  - weekly temperature compared to daily temperature
- Weather_Impression
  - used weathercode to classify weather into categories

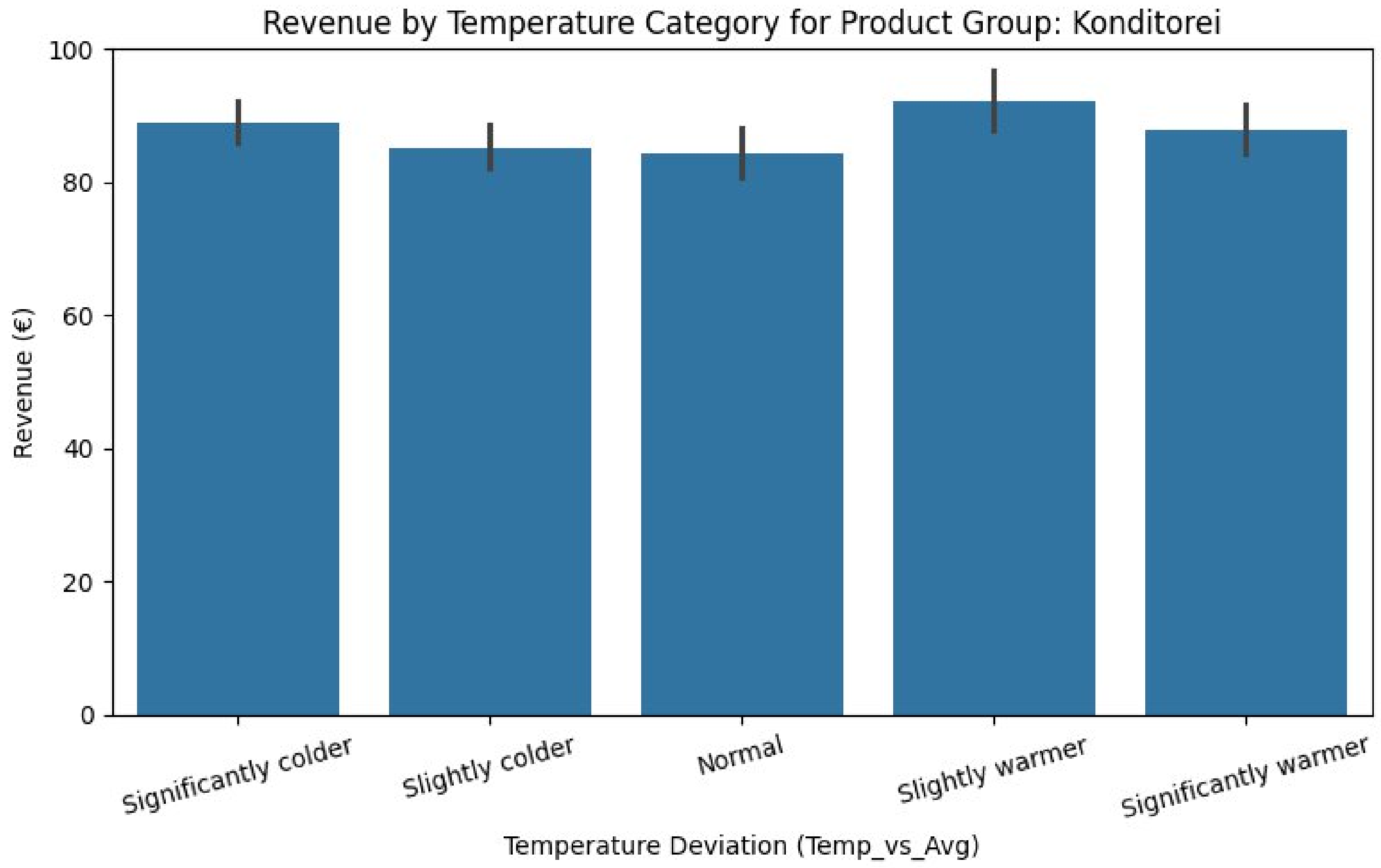
Categorized temperature



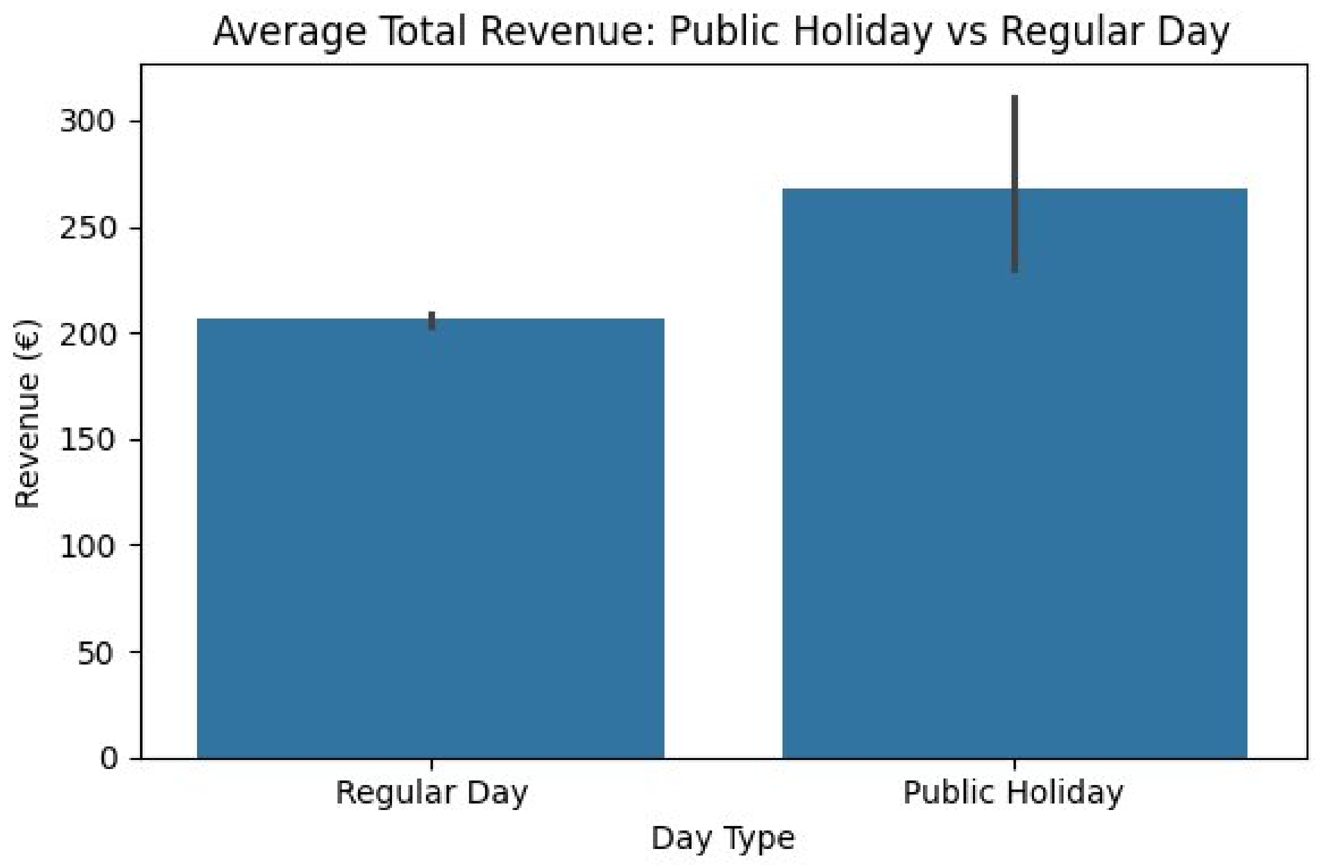
Categorized temperature



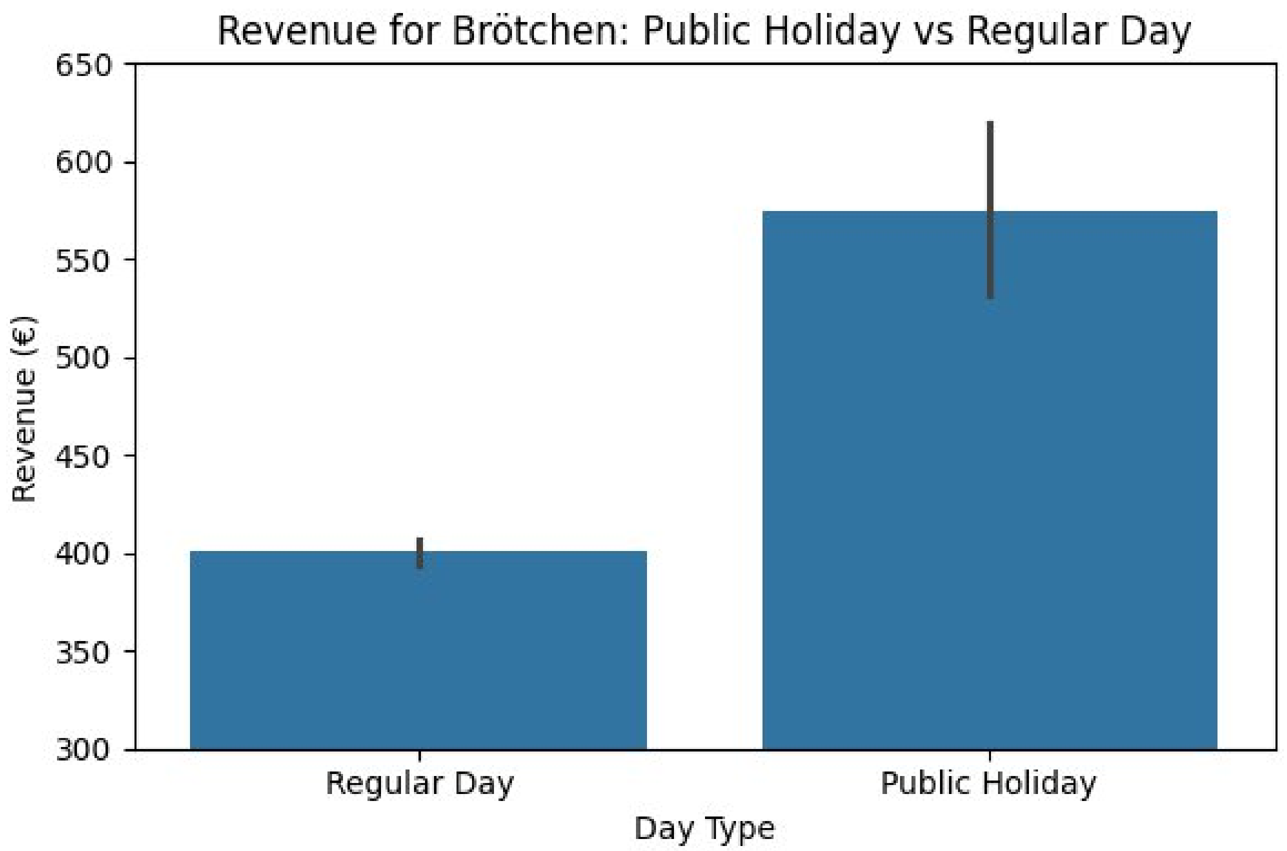
Categorized temperature



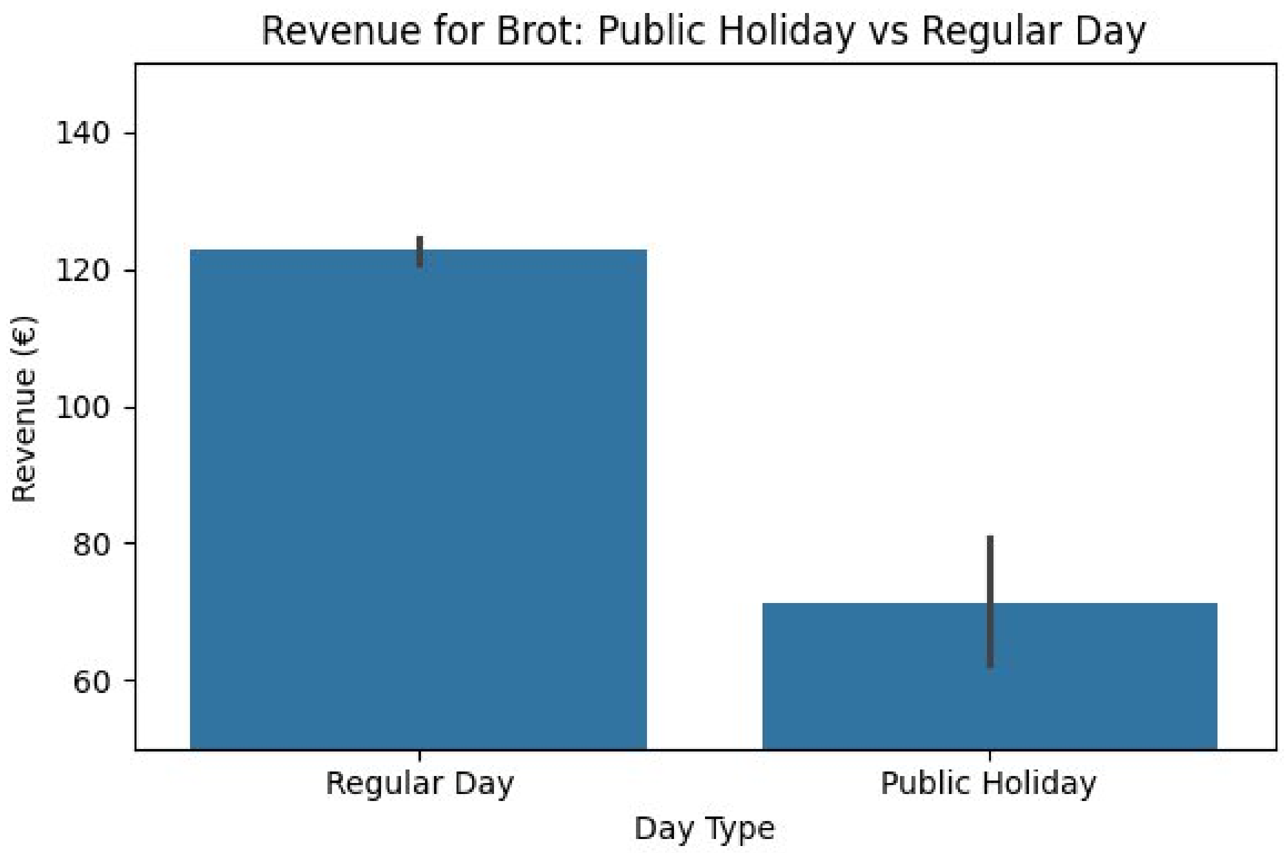
Public Holiday vs Regular Day



Public Holiday vs Regular Day



Public Holiday vs Regular Day

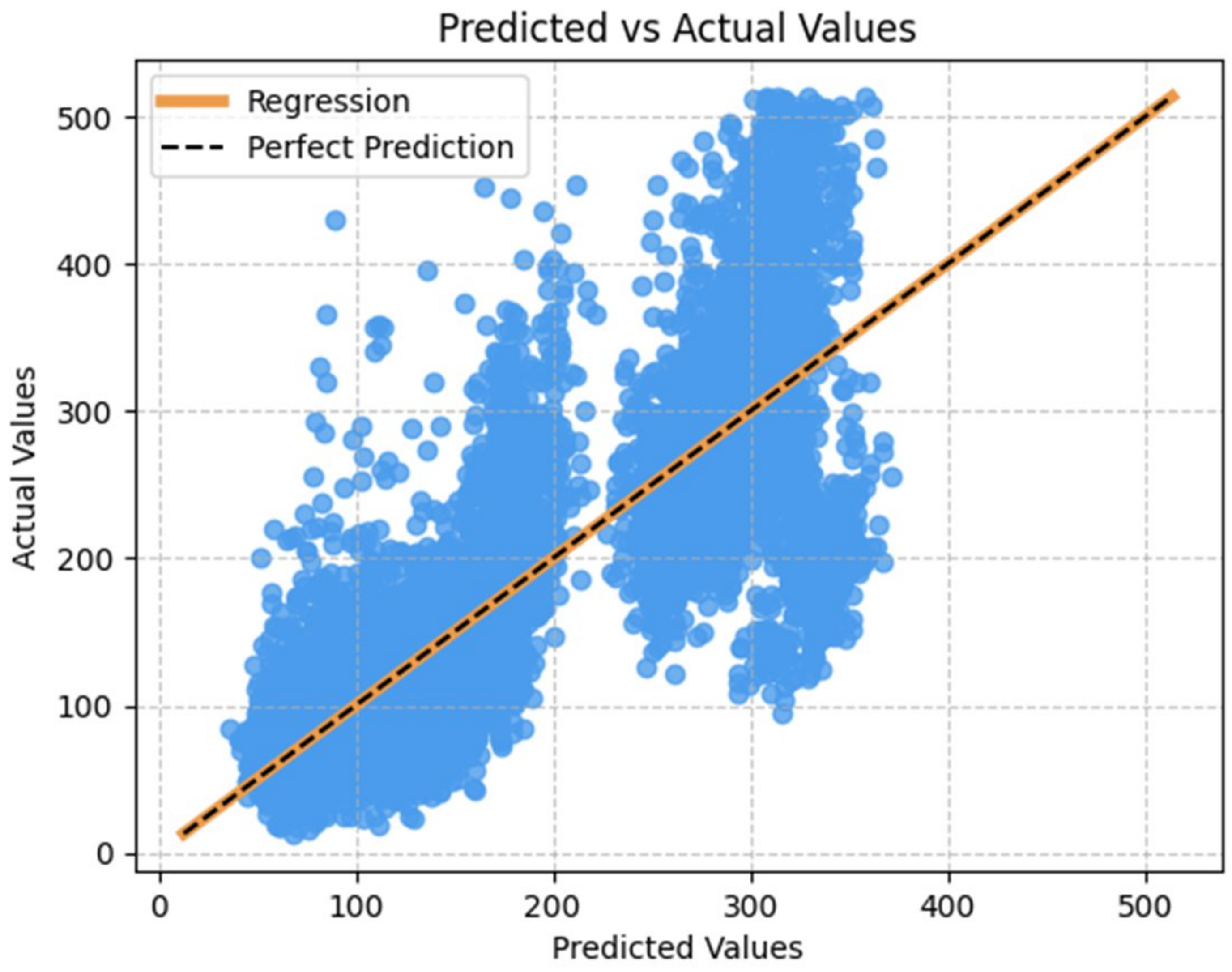


Linear Model - Preparation

- Missing values replaced with the average of the previous and following value
- IQR method: statistical technique for detecting outliers
- IQR = Q3 - Q1
- Q1 (first quartile): 25% of the data is below this value
- Q3 (third quartile): 75% of the data is below this value
- IQR: The range between Q1 and Q3, containing the middle 50% of the data

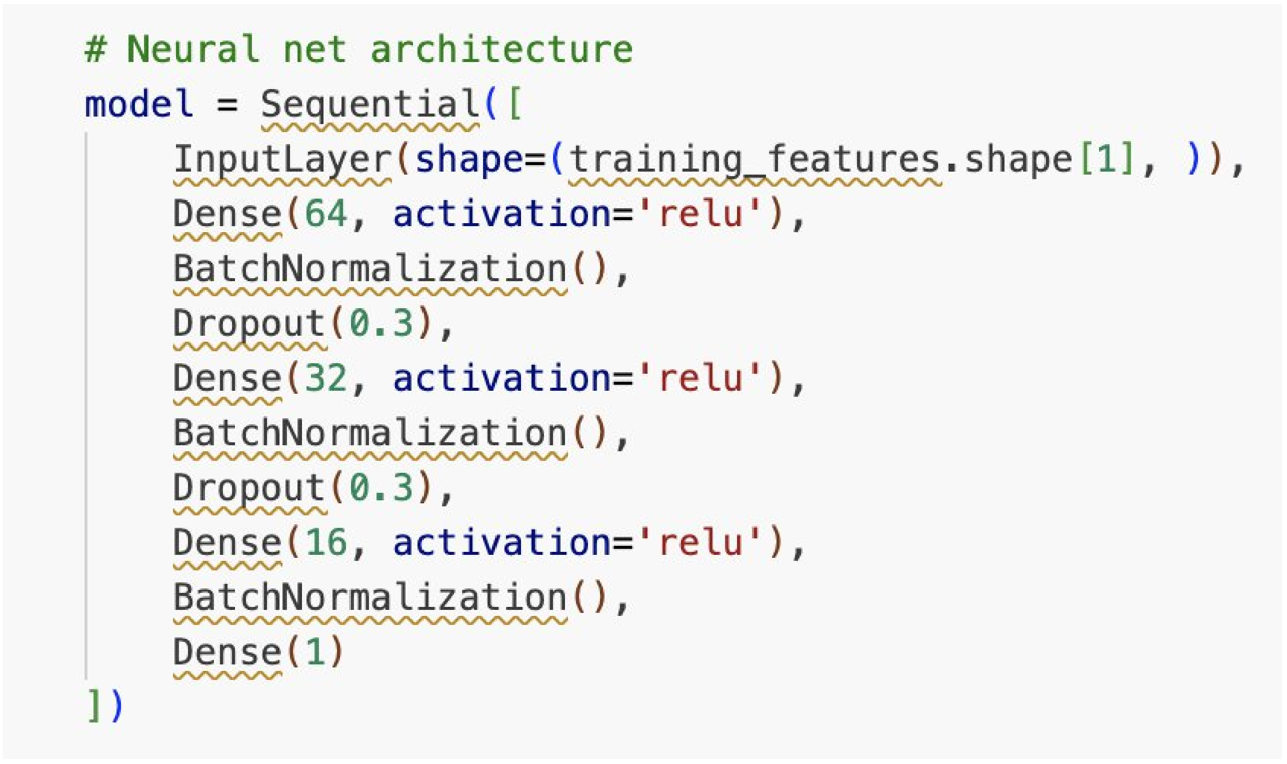
Linear Model

- R² = 0.672
- Kaggle Submission Score: 0.26734

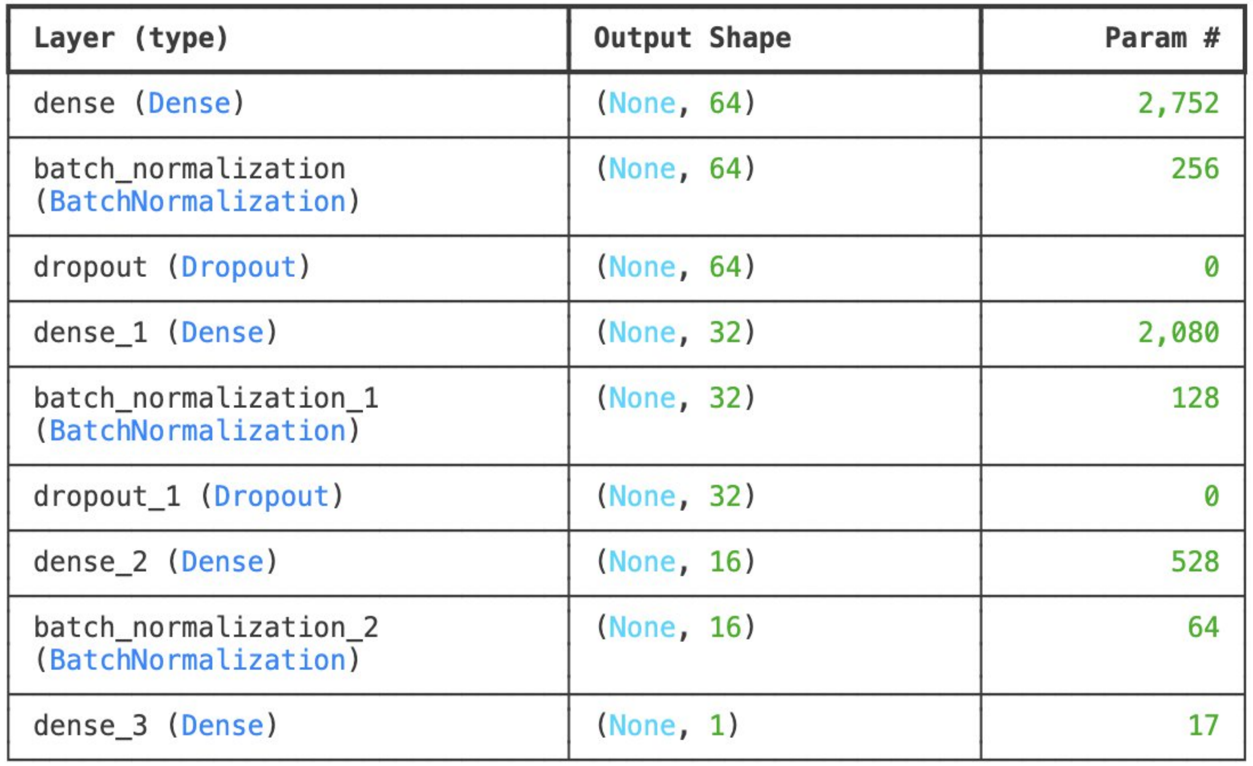


Neural network optimization

- Source code



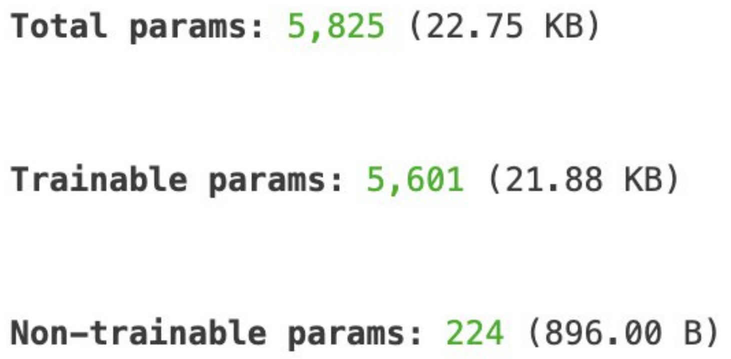
```
# Neural net architecture
model = Sequential([
    InputLayer(shape=(training_features.shape[1], )),
    Dense(64, activation='relu'),
    BatchNormalization(),
    Dropout(0.3),
    Dense(32, activation='relu'),
    BatchNormalization(),
    Dropout(0.3),
    Dense(16, activation='relu'),
    BatchNormalization(),
    Dense(1)
])
```



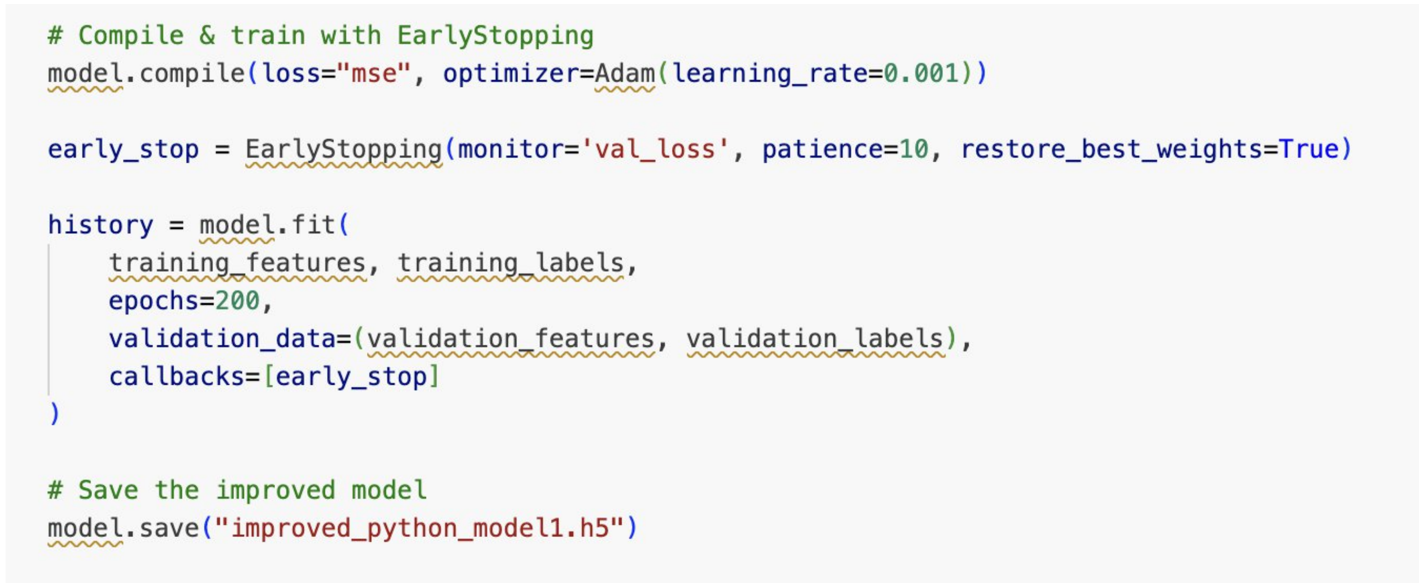
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	2,752
batch_normalization (BatchNormalization)	(None, 64)	256
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
batch_normalization_1 (BatchNormalization)	(None, 32)	128
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 16)	528
batch_normalization_2 (BatchNormalization)	(None, 16)	64
dense_3 (Dense)	(None, 1)	17

Neural network optimization

- Source code



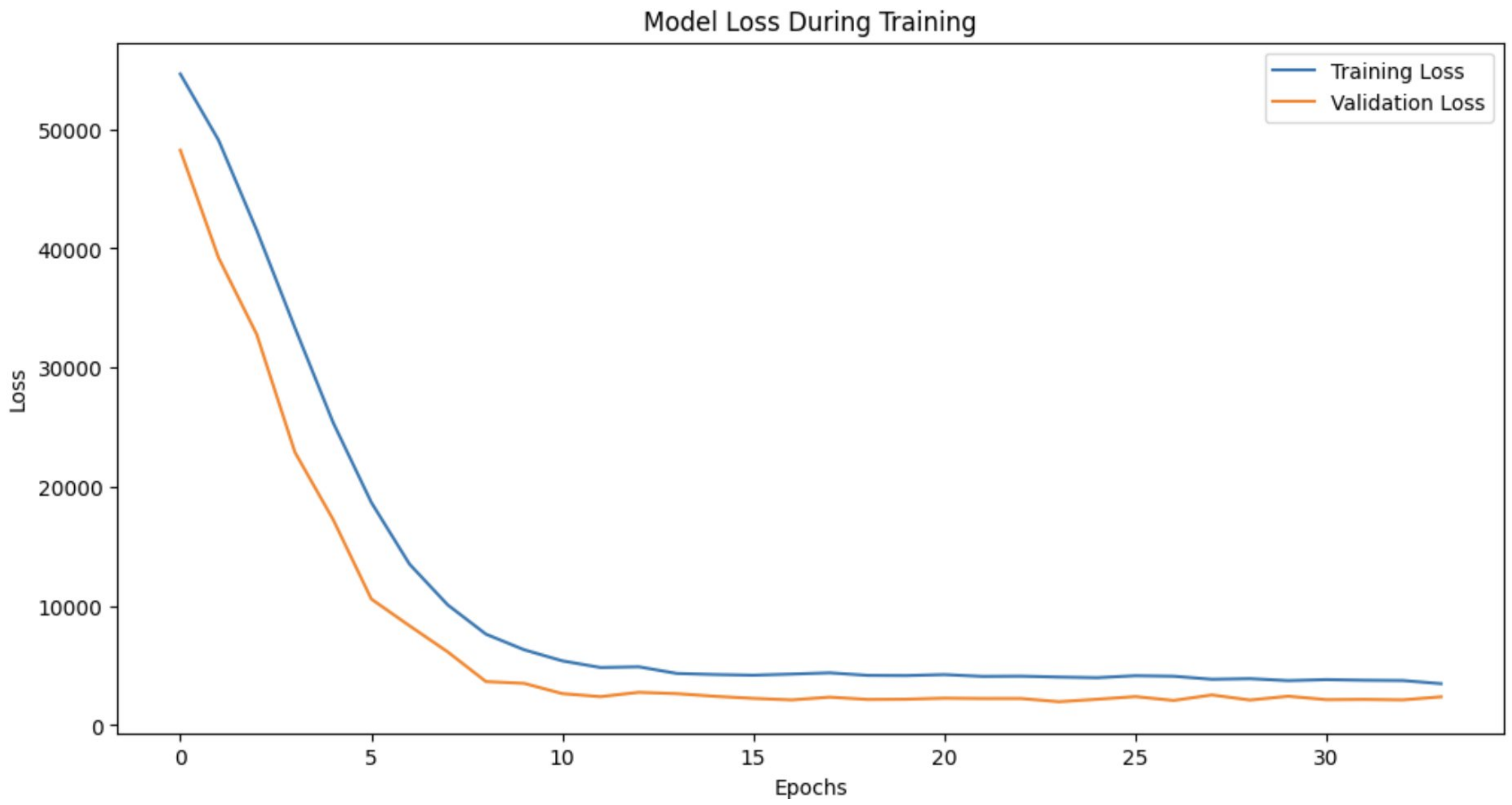
```
Total params: 5,825 (22.75 KB)

Trainable params: 5,601 (21.88 KB)

Non-trainable params: 224 (896.00 B)
```



```
# Compile & train with EarlyStopping
model.compile(loss="mse", optimizer=Adam(learning_rate=0.001))

early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

history = model.fit(
    training_features, training_labels,
    epochs=200,
    validation_data=(validation_features, validation_labels),
    callbacks=[early_stop]
)

# Save the improved model
model.save("improved_python_model1.h5")
```

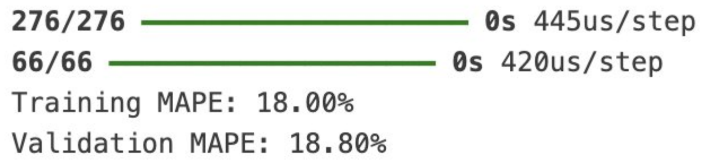
Neural network optimization

- Loss function plot

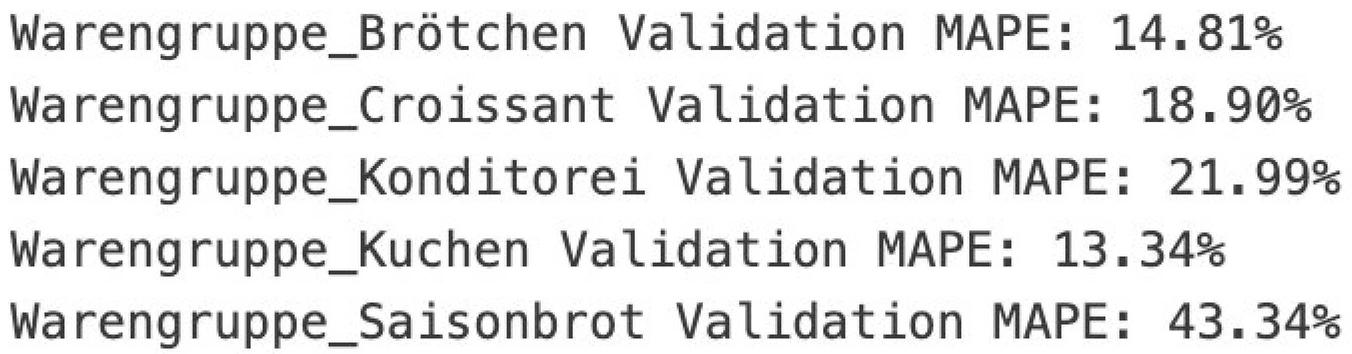


Neural network optimization

- MAPE



```
276/276 ━━━━━━━━━━━━━━━━━━━━ 0s 445us/step
66/66 ━━━━━━━━━━━━━━━━━━━━ 0s 420us/step
Training MAPE: 18.00%
Validation MAPE: 18.80%
```



```
Warengruppe_Brötchen Validation MAPE: 14.81%
Warengruppe_Croissant Validation MAPE: 18.90%
Warengruppe_Konditorei Validation MAPE: 21.99%
Warengruppe_Kuchen Validation MAPE: 13.34%
Warengruppe_Saisonbrot Validation MAPE: 43.34%
```

Highlights & Lowlights

Worst Fail

- Weather data imputation
  - Getting weather data from weather station

- Adding time-series features
  - Took a lot of times, didn't improve the model

Best Improvement

- Keeping it simple
  - Resulted in best MAPE and R^2