Splitting Datasets

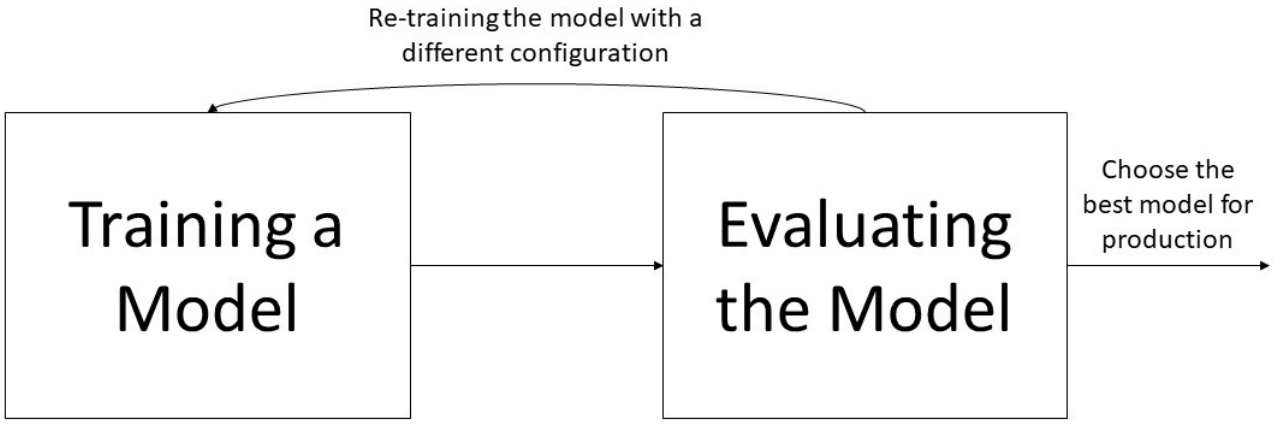
Split the overall project's dataset into training and evaluation sets.

Chapter Goals:

- Learn about training and evaluation sets
- Split the project's final dataset into training and evaluation sets

A. Training and evaluation

There are two main components in creating a machine learning model: training and evaluation. Training is the foundation of machine learning, but evaluation is just as important. Model evaluation gives us a concrete idea of just how good the model is after training, and it allows us to compare the performances for different configurations of the model.



The process for training and evaluating a machine learning model.

B. Set proportions

The question now becomes how much of the data we use for training and how much we use for evaluation. We should use a lot more data for training (the training set) compared to the data for evaluation (the evaluation set). The exact amount is up to the machine learning engineer to decide.

Using more data in training would potentially improve the model's

performance, but it would limit us in how accurate our evaluation is due to the limited evaluation set size. On the other hand, having a larger evaluation set would give us more confidence in our evaluation process' accuracy, but it might limit the amount and diversity of the data in training.

In our case study, we choose a 90-10 split, meaning that the training set comprises 90% of the final dataset while the evaluation set comprises 10%. Since the overall dataset is pretty large, a 10% evaluation set still gives us a good representation of the overall dataset. Therefore, we can afford to put 90% of the dataset in training to maximize the model's performance.

C. Removing systematic trends

Before splitting the final dataset into training and evaluation sets, we need to randomly shuffle it. This is because the dataset is currently sorted by date and store, which is a systematic trend that we need to remove.

```
print(final_dataset.columns)
print(final_dataset)
```

The final_dataset object from the Preliminary Data Analysis section, with the 'Date' feature not dropped. It is currently sorted by the 'Date' and 'Store' features.

If we don't remove this trend, each training step will have data that is too similar to adjacent training steps, since they will likely be for adjacent weeks of the same store. This is an artificial characteristic that doesn't appear in real life, so it would negatively impact the model for real life predictions.

```
# Split the final pandas DataFrame into training and evaluation sets
def split_train_eval(final_dataset):
    # CODE HERE
    pass
```