Optimizing the Dataset for Training

Training is often resource-intensive. Either memory, CPU, or GPU limits may prevent you from completing the training step. While there are several techniques you can use to reduce dataset size, one of the more common ways is by just sampling the data. Others include batch processing and distributed training.

```
import pandas as pd

df = pd.DataFrame([
    ["cat", 1.0, "3-2021"],
    ["cat", 0.5, "1-2021"],
    ["dog", 0.2, "5-2021"],
    ["bird", 3.3, "3-2021"]])

# returns sample of df with size of 2
df.sample(n=2, random_state=0)


# output
      0    1       2
2   dog  0.2  5-2021
3  bird  3.3  3-2021
```

Scikit-learn Model Training API

- Every algorithm has parameters that change how the model behaves
- Parameters change training functionality as well, such as duration length
- Scikit-learn has default values if not manually set

```
import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.DataFrame([[5, 3.4, 6], [1, 0.4, 10], [2, 0.1, 1]])
target = [0, 1, 1]

# One line model creation
reg = LinearRegression().fit(df, target)

# Score model with default metrics
print(reg.score(df, target))

#output
1.0

# Predict targets
print(reg.predict(df))

# output
[-2.22044605e-16  1.00000000e+00  1.00000000e+00]
```

Additional Resources

- Scikit-learn has great resources about which models are available: **User Guide**