Exercise: Dataset Principles

Since models are nothing without data, it's important to make sure the fundamentals are strong when creating and shaping your datasets. Here we'll create a regression dataset and split it into the three core dataset types: train, validation, and test.

Your tasks for this exercise are:

1. Create a dataframe with your features and target arrays from `make_regression` .
2. Create a 60% Train / 20% Validation / 20% Test dataset group using the `train_test_split` method.
3. Confirm the datasets are the correct size by outputing their shape.
4. Save the three datasets to CSV

In [1]:

```
import pandas as pd
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
```

In [7]:

```
# Creating a regression dataset with 1000 samples, 5 feature columns, 2 which are actually useful, and 1 target column
regression_dataset = make_regression(
    n_samples=1000, n_features=5, n_informative=2, n_targets=1, random_state=0
)
```

In [9]:

```
df = pd.DataFrame(regression_dataset[0])
df["target"] = regression_dataset[1]
```

In [10]:

```
df.head()
```

Out[10]:

	0	1	2	3	4	target
0	0.236225	-0.323289	-0.018429	-1.548471	1.311427	70.618083
1	-0.801497	0.271170	-0.525641	-0.887780	0.936399	52.757870
2	0.687881	0.417044	-1.203735	0.498727	-0.737932	-43.728456
3	-0.679593	-1.063433	-1.797456	0.913202	2.211304	156.835125
4	0.096479	-0.507060	0.522083	0.155794	1.520004	102.748706

In [14]:

```
# Create a train: 0.8 | test: 0.2 ratio dataset
df_train, df_test = train_test_split(df, test_size=0.2, random_state=0)

# Create a train: 0.6 | validation: 0.2 ratio dataset
df_train, df_val = train_test_split(df_train, test_size=0.25, random_state=0)

# Final dataset sizes: train: 0.6, validation: 0.2, test: 0.2,
```

In [15]:

```
# Output each shape to confirm the size of train/validation/test
print(f"Train: {df_train.shape}")
print(f"Validation: {df_val.shape}")
print(f"Test: {df_test.shape}")
```

```
Train: (600, 6)
Validation: (200, 6)
Test: (200, 6)
```

In [18]:

```
# Output all datasets to csv
df_train.to_csv(("train.csv"), index=False)
df_val.to_csv("validation.csv", index=False)
df_test.to_csv(("test.csv"), index=False)
```