03_exercise_solution

November 9, 2021

1 Exercise: Model Training and Evaluation

Now that we have the data fundamentals for creating, cleaning, and modifying our datasets, we can train and evaluate a model, in this case it's a linear regression model.

Your tasks for this exercise are: 1. Create a dataframe with the regression dataset, include the features and target within the same dataframe. 2. Create a 60% Train / 20% Validation / 20% Test dataset group using the `train_test_split` method. 3. Fit the LinearRegression model on the training set. 4. Evaluate the model on the validation set. 5. Evaluate the model on the test set.

```
In [25]: import pandas as pd
         from sklearn.datasets import make_regression
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
```

```
In [26]: regression_dataset = make_regression(
             n_samples=10000,
             n_features=10,
             n_informative=5,
             bias=0,
             noise=40,
             n_targets=1,
             random_state=0,
         )
```

```
In [27]: # Create the dataframe using the dataset
         df = pd.DataFrame(regression_dataset[0])
         df["target"] = regression_dataset[1]
```

```
In [28]: # `.head()` to view what the dataset looks like
         df.head()
```

```
Out[28]:           0         1         2         3         4         5         6  \
         0 -1.039309 -0.533254  0.006352 -0.130216 -0.672371 -1.227693 -1.605115
         1  0.906268  1.112101 -0.816500  0.461619  0.883569  1.125719 -0.993897
         2  0.334137  0.320004 -0.248267 -0.317444  0.834343  1.381073  0.901058
         3  0.250441 -1.215110 -1.562450  0.162566 -1.630155 -0.449801 -1.033361
         4 -1.440993 -0.388298 -0.431737  0.518420 -0.405904 -0.785488  1.008090
```

```
          7         8         9      target
0  0.313087  1.709311  1.486217 -190.336109
1  0.999854 -1.919401 -1.137031   33.264389
2 -0.655725  0.340868 -1.481551  120.287805
3 -0.671750 -1.331549 -0.979638 -472.599566
4 -0.695019  1.885108 -0.913755   42.355214
```

```
In [29]: # train: 0.8 | test: 0.2
         df_train, df_test = train_test_split(df, test_size=0.2, random_state=0)

         # train: 0.6 | validation: 0.2
         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, text: 0.2,
```

```
In [30]: # 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: (6000, 11)
Validation: (2000, 11)
Test: (2000, 11)
```

```
In [31]: # Train the linear model by fitting it on the dataframe features and dataframe target
         reg = LinearRegression().fit(df_train[range(10)], df_train["target"])
```

```
In [32]: # Evaluate the linear model by scoring it, by default it's the metric r2.
         reg.score(df_val[range(10)], df_val["target"])
```

```
Out[32]: 0.9349344900971387
```

```
In [33]: # Once done optimizing the model using the validation dataset,
         # Evaluate the linear model by scoring it on the test dataset.
         reg.score(df_test[range(10)], df_test["target"])
```

```
Out[33]: 0.9323863267980969
```