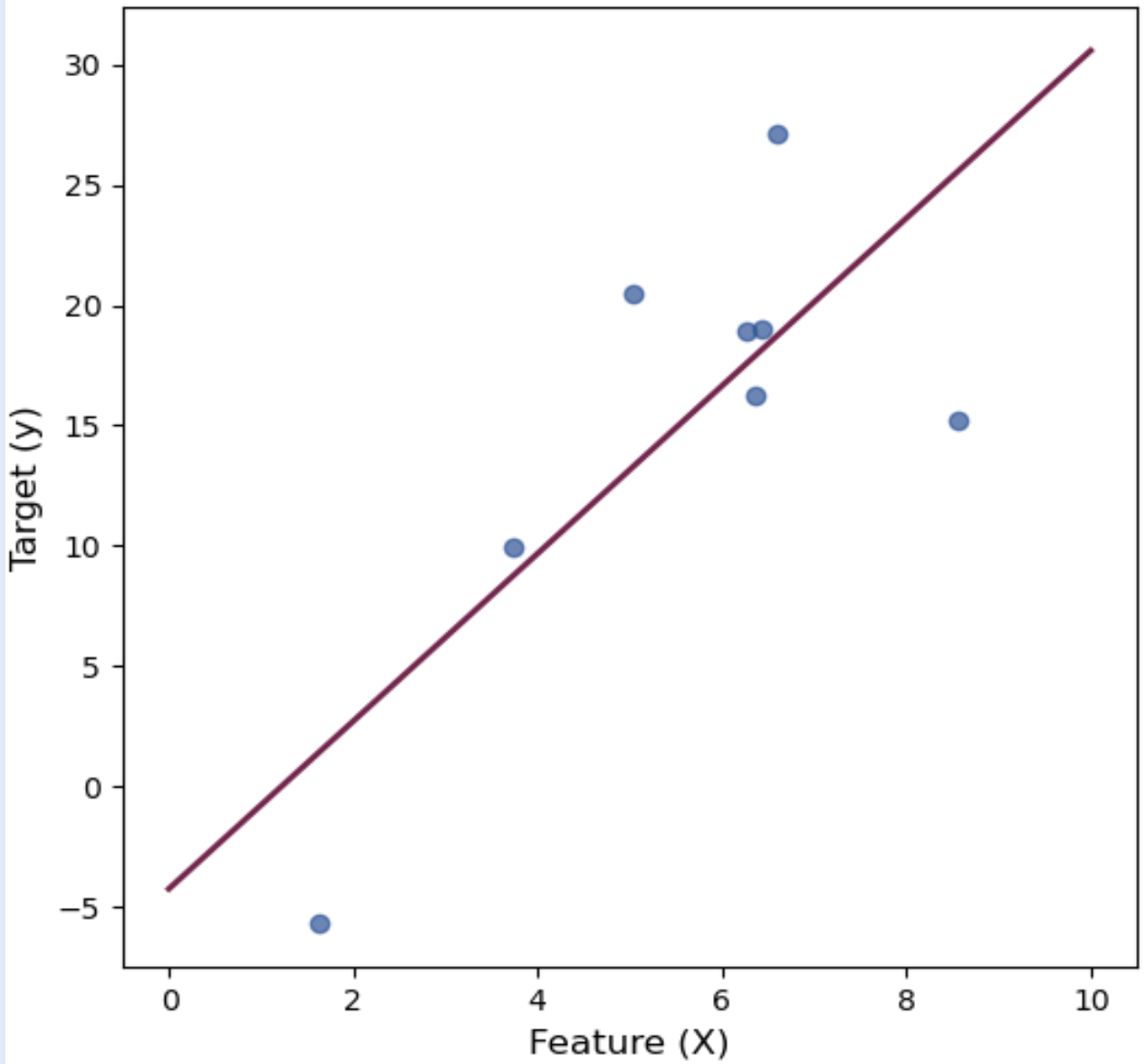
Cost Functions

- MSE
- MAE
- RMSE

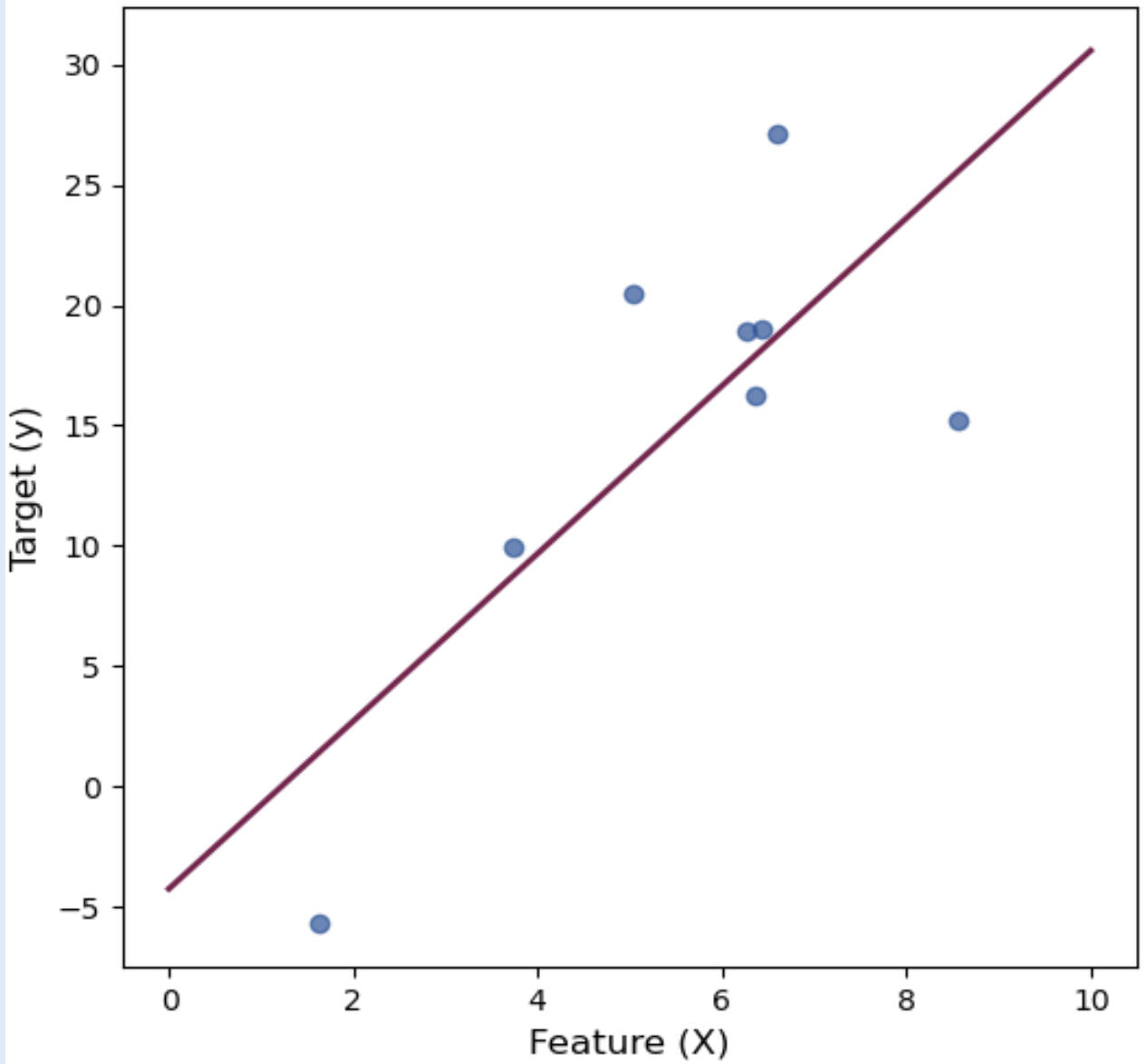
Mean Squared Error (MSE) is a good choice when you want to emphasize larger errors by penalizing them more heavily. However, it's less interpretable since it isn't in the same units as the dependent variable, like MPG.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y} - y)^2$$



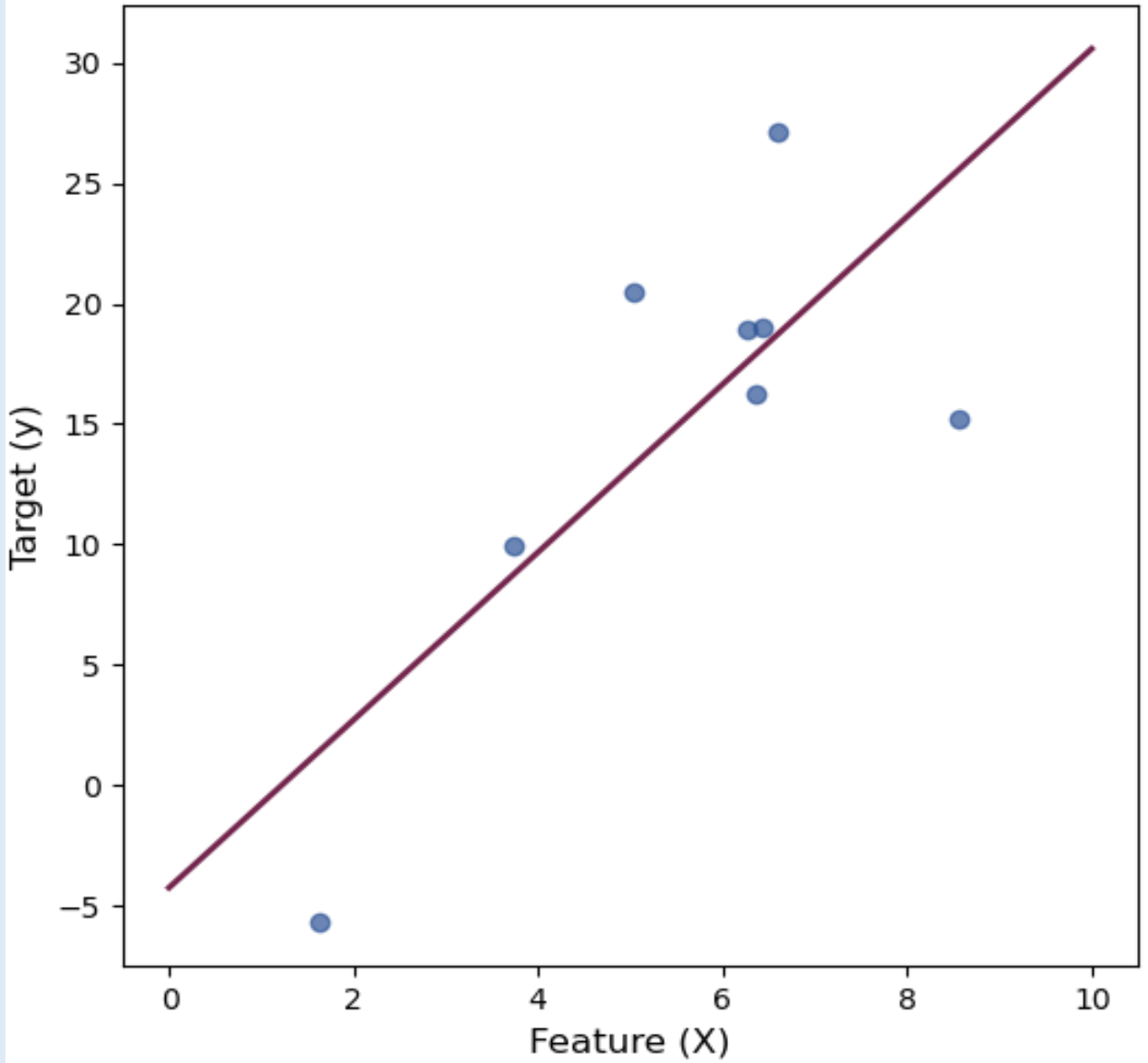
Mean Absolute Error (MAE) is an excellent choice when you need a straightforward, easy-to-interpret metric that represents the average error in the same units as the dependent variable, such as MPG.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y} - y|$$



Root Mean Squared Error (RMSE) offers the best of both worlds, balancing sensitivity to large errors with interpretability, as it's expressed in the same units as the dependent variable, such as MPG.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y} - y)^2}$$



Import the appropriate libraries

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

Load the data set

Process the data set (for example handle missing data and problematic data types)

```
x= x.reshape(-1,1)
y= y.reshape(-1,1)
```

Reshape the data if needed,

Split the data to testing and training sets

```
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.2, random_state=42)
```

```
model = LinearRegression()

model.fit(x_train, y_train)
```

Choose the model

Fit the model with data (=train it)

```
y_pred = model.predict(x_test)
```

Make a prediction

```
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
```

Test your model

Analyze the result