Linear Regression with scikit-learn

Linear Regression

scikit-learn: Using Linear Algebra

```python
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()
lin_reg.fit(X, y)
lin_reg.coef_, lin_reg.intercept_
```

Note: this is not a learning algorithm.

Optimizers

1. (Batch) Gradient Descent
2. Stochastic Gradient Descent
3. Mini-Batch Gradient Descent

scikit-learn: Stochastic Gradient Descent

```python
from sklearn.linear_model import SGDRegressor

sgd_reg = SGDRegressor()
sgd_reg.fit(X, y)
sgd_reg.coef_, sgd_reg.intercept_
```

scikit-learn: SGD Hyperparameters

```python
from sklearn.linear_model import SGDRegressor

sgd_reg = SGDRegressor(
    max_iter=100000,
    n_iter_no_change=10,
    tol=1e-4,
    learning_rate='adaptive',
)
sgd_reg.fit(X, y)
sgd_reg.coef_, sgd_reg.intercept_
```

scikit-learn: SGD `partial_fit`

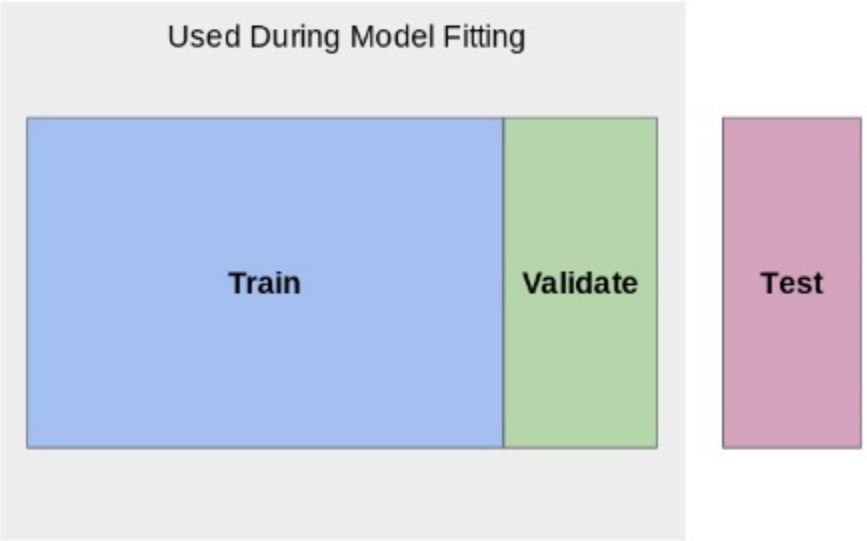
```python
from sklearn.linear_model import SGDRegressor

sgd_reg = SGDRegressor()
sgd_reg.partial_fit(X_1, y_1)
sgd_reg.partial_fit(X_2, y_2)
...
sgd_reg.partial_fit(X_n, y_n)
sgd_reg.coef_, sgd_reg.intercept_
```

Loss

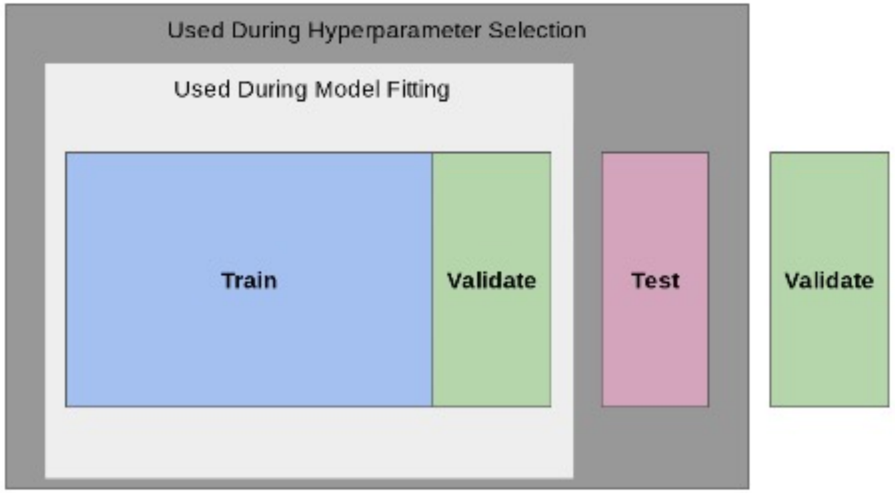
Mean Squared Error

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

Train/Validate, Test



Train/Validate, Test, Validate


Used During Hyperparameter Selection
Used During Model Fitting
Train
Validate
Test
Validate

Your Turn