Gradient Descent in Scikit-Learn: Takeaways

Syntax

CAPTURE GROUPS

- Importing `SGDClassifier` and `SGDRegressor`:

```
from sklearn.linear_model import SGDClassifier, SGDRegressor
```

- Importing `classification_report`:

```
from sklearn.metrics import classification_report
```

- Instantiating `SGDRegressor` with a few of the most important parameters:

```
SGDRegressor(loss=loss_function, learning_rate="constant", eta0=learning_rate, tol=tolerance, max_iter=iterations)
```

- Instantiating `SGDClassifier` with mean squared error for a loss function and balancing an imbalanced dataset:

```
SGDClassifier(loss="squared_error", class_weight="balanced")
```

Concepts

- The standard implementation of gradient descent can be too slow when dealing with multiple predictors because of increased minima and ground to cover.
- Stochastic gradient descent alleviates these problems by using just a sample of the data for the updates to weight and bias.
- Scikit-learn's implementation for stochastic gradient descent also updates the learning rate.
- Gradient descent can also be used for classification.

Resources

- Precision and recall
- F-score
- Classification report