Turbo Boost

It's time to power up your decision-tree machine-learning algorithm with a **boosting algorithm**. Using a **boosting** algorithm will result in a more robust algorithm with a better ability to classify fraudulent transactions.

Instructions

1. Open the starter file, and navigate to the **Choose Optimal Learning Rate** section. Choose the learning rate that produced the best model accuracy. Use this rate to build your own `GradientBoostingClassifier`.
2. Instantiate a `GradientBoostingClassifier` using the **100** `n_estimators`, the learning rate from step 1 as `learning_rate`, **5** `max_features`, `max_depth` of **3**, and `random_state` of **0**.
3. Use the `X_train_scaled` and `y_train` datasets, as well as the **sklearn** `fit` function, to fit the model. Note: Use the `ravel` function when using the `y_train` dataset.
4. Score the model using the **sklearn** `score` function.
5. Use the `predict` function to make predictions.
6. Use the `accuracy_score` function to check the accuracy of the predictions. Hint: `accuracy_score(y_test, predictions)`.
7. Generate a **confusion matrix** and **classification report**.
8. Evaluate the model using the **confusion matrix** and **classification report**. Did the model perform as you expected? How did it execute compared to the other decision-tree algorithms?