Supervised : Advanced methods

1 Fundamentals

1. Preparing the dataset
   (a) Get ready to use the same dataset as for the previous class. Keep columns [budget, popularity, revenue, runtime, vote_average, vote_count] as numerical columns. Convert 0 in budget, revenue, runtime and vote_count into na.
   (b) Add year, month and Adult as in a previous class.
   (c) Standardize the data
2. Regularization
   (a) Train first a classic linear regression. Plot its performance (at least MAE and R2) and its coefficients.
   (b) Train a `Lasso` regressor, first with the default `alpha` parameter of 1. Plot its performance, coefficients, and compare with non regularized result.
   (c) Do the same with a `Ridge` regressor. You should not observe large change compared with the unregularized model, for the default parameter.
   (d) To see the effect of parameters on coefficients, make the parameter vary from 1 to 10 in log scale, and plot the evoluation of coefficients. You can take inspiration from `https://scikit-learn.org/stable/auto_examples/linear_model/plot_ridge_path.html`
   (e) Do the same with Lasso, with the appropriate range of parameter.
3. XGboost
   (a) As a baseline, train a Decision Tree, with a quick manual calibration of parameters to get a decent result
   (b) XGBoost is provided in an independent library called `xgboost`. However, it provides a sklearn compatible interface. Check `https://xgboost.readthedocs.io/en/stable/python/python_api.html#module-xgboost.sklearn` for its reference
   (c) Train XGBoost with default parameters on your data. Compare results with other methods.

2 Advanced

(a) You have trained 3 different models. Would combining them improve the results? Let's do manually some bagging.
(b) Start by splitting your dataset in 3, train, validation, test.
(c) Train each of your models on the train.
(d) Build a new dataframe dfTemp in which the columns correspond to the `predict` of the values of your models on the validation dataset.
(e) Train a new model of your choice, that use as dfTemp as X, with the real Y of the validation set as target. This model learns how to combine the predictions of the other.
(f) Compute the performance of your model on the test set. Compare with the performance of each of the individual models on the test set.