Supervised : basics

The objective of those exercises is to practice the basics of supervised machine learning.

1 Fundamentals

1. Preparing the dataset
   (a) Load the `cars_synth_clean.csv` dataset.
   (b) Split your dataset into 1)a train and test set, 2) A dataset of features used for prediction (X) and a target variable (Y), the popularity. You can use sklearn `train_test_split` function. Keep for instance 1/3 as test set. Use random samples (Check the function's parameters)
2. Linear Regressions
   (a) Train a linear regression using only the column `length` to predict the `price`. You should use the `LinearRegression` class from sklearn library. Use method `fit` to train the model, and `predict` to get prediction. Note: sklearn functions accept whole dataframes, but not columns/Series.
   (b) Compute the scores we have seen (MSE, RMSE, MAE, R2), using corresponding functions in sklearn. Do it first by using the train set, and then using the test set. Compare the difference. How to interpret the MAE? Would you say it is a useful prediction? How to interpret the R2? Would you say it is a good prediction?
   (c) Re-run the code including the random samples several times. Check that the values change significantly.
   (d) To get a more intuitive idea of the performance, plot the relation between the target variable and the prediction (e.g., seaborn scatterplot, x=target variable, y=your prediction). With a perfect prediction, you should observe a diagonal line.
   (e) Let's now use all the numeric values at our disposal. Check how much it improves the results.
3. Feature Engineering
   (a) Make a scatterplot showing the relation between `age` and `year`. Observe that there is a non-linear relationship. Find a way to make it mostly linear (you will need to use a log, but also remember the ratio/interval distinction...).
   (b) Create a new column "Age" with the transformed column having a linear relation with the target
   (c) Check how much the performance improve using that new column
   (d) Transform all the categorical columns into multiple boolean variables. You can use pandas `get_dummies` (or with sklearn `OneHotEncoder`, less convenient).
   (e) Check how much the performance improved using all information
   (f) You can now interpret the increase in performance in term of MAE, R2, and using a scatter plot between ground truth and prediction
4. Decision tree
   (a) Train a Decision tree (`DecisionTreeRegressor`) with sklearn, using default parameters, on the original dataset without transformations

(b) First, evaluate on the training set, then on the test set. You should observe massive overfit.

(c) The default parameters are bad. Play with the parameters `max_depth`, `min_samples_leaf`, `max_leaf_nodes`, to try to limit the overfit (you should be able to improve over the linear regression)

(d) Train a tree small enough to be visualized, and plot it. You can use the built-in tool following the documentation `https://scikit-learn.org/stable/modules/tree.html#tree`. The `graphviz` method usually gives the nicest results.

(e) Try to interpret the tree.

2 Going Further

(a) Try to make the best prediction possible using your knowledge.

(b) Do the same exercise on the real dataset of used cars