Machine Learning Pipeline Assignment: Wine Quality Dataset

Data Science Course

September 2024

Assignment Week 2

The use of Jupyter notebooks from e.g. Anaconda is recommended for those who are not fluent in VScode.

1. Load the white wine dataset from the UCI Machine Learning Repository associated with the paper by Cortez et al., *Modeling wine preferences by data mining from physicochemical properties*.
2. Split off a test set of 20% (stratification is not necessary).
3. Train a KNN classifier with 1 neighbor on the training set and make predictions on the test set.
4. Calculate the Mean Absolute Error (MAE) for the predictions (Note: MAD in the paper is our MAE)
5. Compare your model's performance with Table 2 from the paper. How does your model perform?
6. Apply a standard scaler by fit_transforming the training set and transforming the test set.
7. Use these scaled sets to train a KNN classifier with 1 neighbor on the transformed training set and make predictions on the transformed test set.
8. Calculate Mean Absolute Error (MAE) for the predictions.
9. Compare your results with Table 2 and your earlier model. How does your model perform?
10. Create a pipeline with the standard scaler and repeat the previous step. Do you get the same result?
11. In the article, they perform 5-fold cross-validation (no stratification). Do the same using a pipeline with scaling.

12. Provide the Mean Absolute Error (MAE) averaged over the validation sets, including the uncertainty margin due to the differences between the validation sets.

13. Compute the confusion matrix (use the combined predictions from the validation sets).

14. Now, do the same with a KNN regressor with 1 neighbor (include it directly in the pipeline). What is the difference in performance (MAE and RMSE) compared to the KNN classifier? Could you have expected this?

15. Repeat the comparison between the KNN classifier and KNN regressor, but now with $k = 10$ neighbors (include it directly in the pipeline). Focus on the MAE. Skip the confusion matrix this time (why?).

16. Round the predictions of the KNN regressor using `np.round`. Create a confusion matrix for the regressor. Compare the results of the KNN regressor and KNN classifier in terms of number of correct predictions, MAE, and RMSE. Explain the differences.

17. Create a custom transformer function that rounds the output and ensures it lies between 0 and 10. Include this in the regressor pipeline using a `TransformedTargetRegressor`. (How this is done is explained in the class, if you have troubles finding out: ask!)

18. Use the same pipeline to build a linear regressor, an SVM regressor, and a random forest regressor.

19. Compare the results with Tables 2 and 3 from the paper. Also consider the RMSE.

20. Return to the KNN regressor: optimize the number of neighbors k using grid search. Ensure that you do not perform this on the test sets. Compare the results with earlier findings.

21. Optional: Feel free to explore more models and optimizations. How far do you get?