Machine Learning for Neuroscience

ML4NS

Slides and notebooks: `https://ml4ns.github.io`

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 5. Ensemble models and Kernel-based models

In lecture 5, we cover the basic principles of ensemble and kernel-based models. In our previous lectures, we made two critical assumptions about the data: that each item we were trying to process, classify, or cluster could be represented by a fixed-size feature vector and that the items were linearly separable. In this lecture, we discuss alternative approaches for items/concepts that are not linearly separable and cannot be best represented as fixed-sized feature vectors (e.g. protein sequences, which can be of variable lengths or evolutionary trees, which have variable shape and size).

We first introduce kernel functions, a process that applies a function to transform the data into a higher dimensional space by which the data items can then be compared/processed. We also provide an overview of the different commonly used kernels, including polynomial, sigmoid and the Gaussian radial basis function. Figure 5.1 illustrates the methodology of a kernel-based model by which to compare/process data.

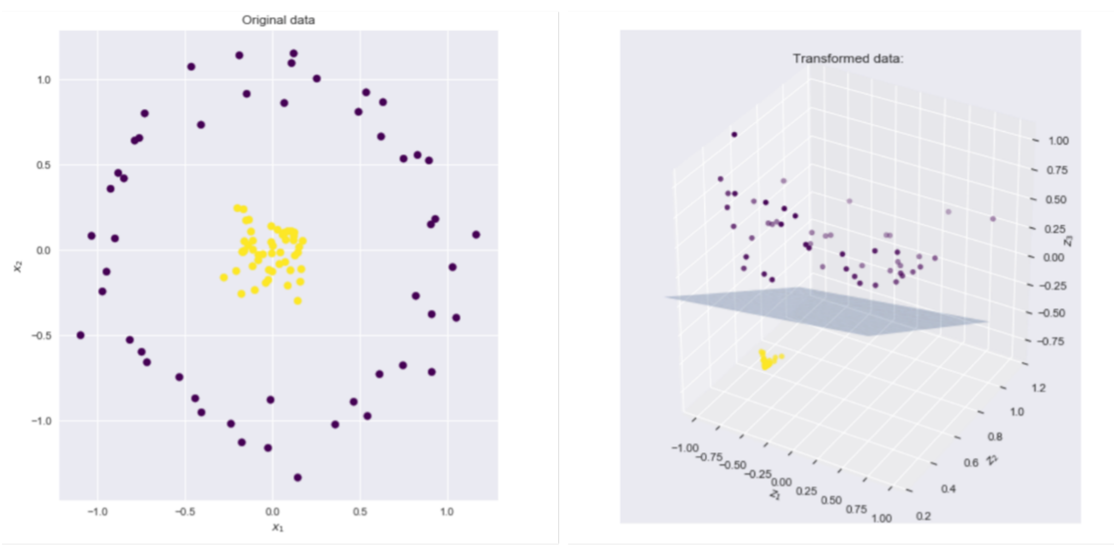

Figure 1. Example of data mapping using a kernel function
Source: `https://xavierbourretsicotte.github.io/Kernel_feature_map.html`

We further expand on how kernel functions can be incorporated into supervised machine learning models using support vector machines, discussing the advantages and disadvantages of such methods.

We then introduce ensemble methods. This is the practice of learning from multiple decision trees. The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalisability/robustness over a single estimator. We also discuss the two main families of ensemble methods: bagging methods in which several base estimators are built independently with their predictions then averaged, and boosting methods, in which base estimators are built sequentially and the aim is to reduce the bias of the combined estimator.

Finally, we introduce the game theoretic approach SHAP (Shapley Additive exPlanations) to explain the output of any machine learning model. We consider how such an approach can be used to connect optimal credit allocation with local explanations regarding how a model makes its decision. Figure 5.2 illustrates a SHAP plot to demonstrate the impact (size and direction of effect) of each feature on the outcome variable.

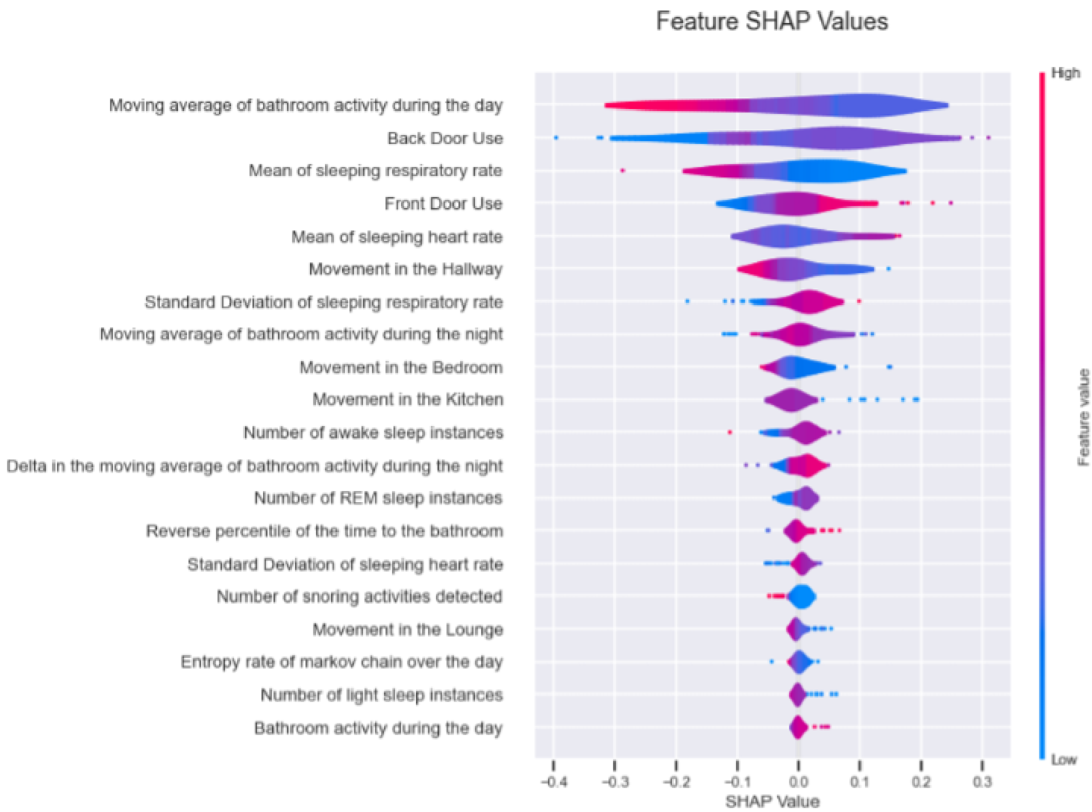

Figure 5.2. An example of a SHAP plot