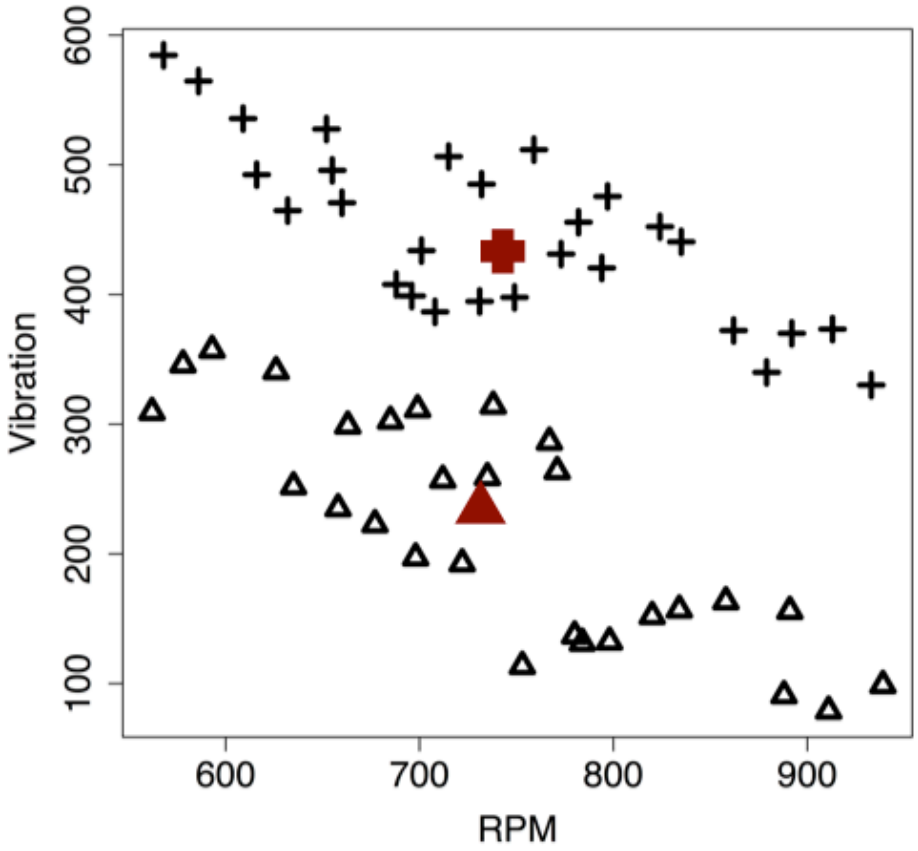

COMP47590

Advanced Machine Learning

Lab Task 2: Implementing Algorithms

Introduction

In this task we want to implement a simple prediction algorithm in scikit-learn. The algorithm we will implement will be a simple template matching predictor, let's call it TemplateMatch. TemplateMatch only works for continuous descriptive features and catectorical target features. TemplateMatch should work very simply as follows:

- **Training:** For each target feature level calculate the average value of all descriptive features for instances that have that target level. Store these average vectors as templates for each target level.
- **Prediction:** When a new prediction needs to be made compare the descriptive feature values of the new query instance to each template and return the target feature level that belongs to the template that is cloesest (based on Euclidean distance) to the query case.

Tasks

Perform the following tasks:

1. Review the documentation on "Rolling Your Own" scikit-learn estimator (or prediction model)[1]
2. Using the sample sciki-learn estimator project write a new estimator class that implements the TemplateMatch algorithm.
3. Evaluate the performance of the TemplateMatch algorithmon the MNIST Fashion dataset.
4. Calcualting distances between query instances and templates is at the core of the TemplateMatch algorithm. The most obvious approach to use for this is Euclidean distance, but ther are plenty of other distance metrics that can be used (e.g. manhattan, chebyshev, and mahalanobis distance[2]). Experiment with alternative distance metrics for the TemplateMatch predictor to see if it improves performance.
5. Can you make the distance metric an option for the TemplateMatch algorithm so that it can be explored using the GridSearchCV scikit-learn object?

[1] http://scikit-learn.org/stable/developers/contributing.html#rolling-your-own-estimator

[2] http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.DistanceMetric.html