Model Selection

COSC5557 – Practical Machine Learning

William Baumchen

12/3/2023

1 - Introduction

When considering solving a problem of some kind using machine learning, once the necessary data has been provided, it becomes important to consider the type of machine learning model used. Each type of machine learning model operates under different principles, and often under different assumptions with respect to the data provided. As such, selecting a model that is optimal can significantly impact the resulting performance. In this report, several different machine learning models were developed and evaluated using the white wine quality data set, and from those models the 'best' was chosen.

2 – Dataset Description

The dataset used is the white wine quality dataset, containing 4,898 observations, with 11 features, all of which are real numbers taken from a continuous range. There is one 'target' feature, with seven possible classes. There are no missing values in the observations and target feature.

3 – Experimental Setup

In this exercise all computation was done utilizing MATLAB, more specifically the Statistics and Machine Learning Toolbox. First, the given white wine quality data was shuffled and split into testing and training sets, that is, a fifth of the overall data set was set aside for later testing purposes. Five different machine learning models were then developed. In this exercise the models used were a linear regression model, a decision tree, a support vector machine regression model, a kernel regression model, and an ensemble regression model. For these models the hyperparameters used were the default values set in the respective MATLAB functions used. For each of the five models a 10-fold cross validation was carried out, such that the resulting trained model could be evaluated without bias. The two evaluations carried out on the models utilized the standard mean-squared error. The linear regression model used in this exercise was the MATLAB fitrlinear function, with default hyperparameters as specified [1]. The decision tree model used in this exercise was the MATLAB fitrtree function, with default hyperparameters as specified [2]. The support vector machine model used in this exercise was the MATLAB fitrsvm function, with default hyperparameters as specified [3]. The kernel regression model used in this exercise was the MATLAB fitrkernel function, with default hyperparameters as specified [4]. The ensemble regression model used in this exercise was the MATLAB fitrensemble function, with default hyperparameters as specified [5].

Once the cross-validation mean squared error and the test mean squared error for each of the models had been found, the results were plotted in boxplots.

4 – Results

After training the different cross-validated models stated above, the resulting cross-validation mean square error was plotted as below in a boxplot in Fig. 1. This is the mean squared error resulting from a randomly partitioned 10-fold cross validation. As can be seen below in Fig. 1, the support vector machine model has the best results, but the mean of its error is worse than the performance of the ensemble learner. The scores found are in Table 1.

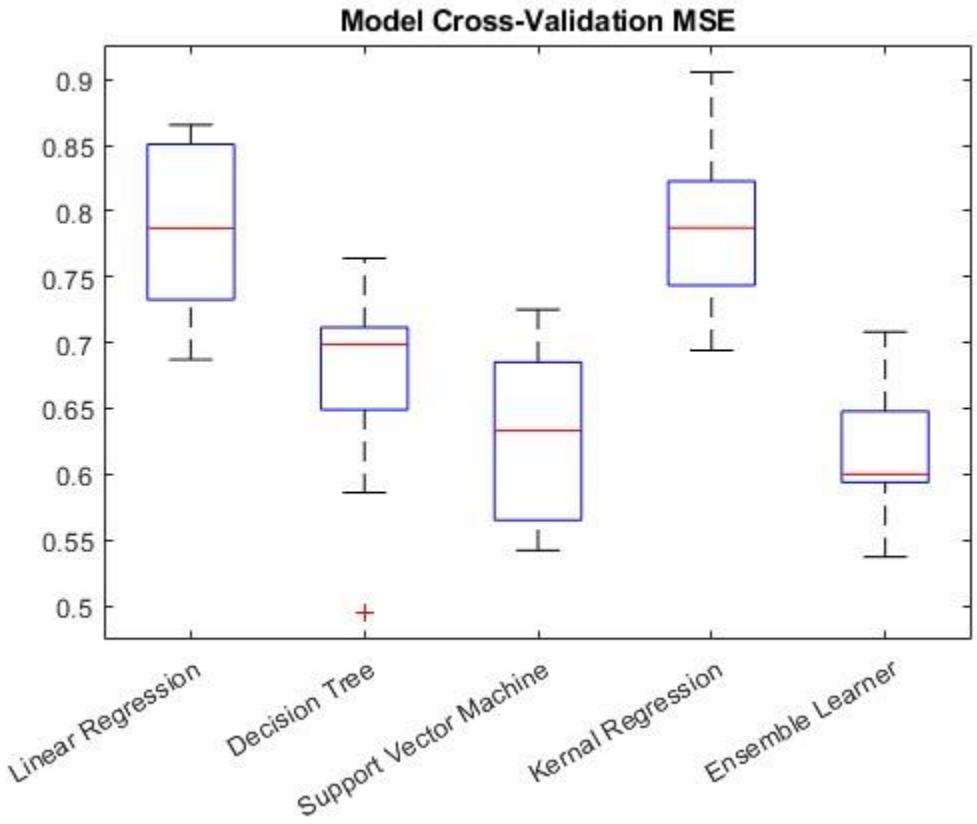

Figure 1 – Model Cross-Validation Mean Squared Error

After completing all training for the different models, each of the individually trained examples of each model type were evaluated using the test dataset, and the resulting mse was plotted in a boxplot below in Fig. 2. Here, the best-performing model type is the support vector machine, possessing both the lowest values for the mean squared error as well as the lowest average mean squared error. These mean error scores are presented in Table 2 below. In addition, the average mean squared error for the different models are shown in Table 3.

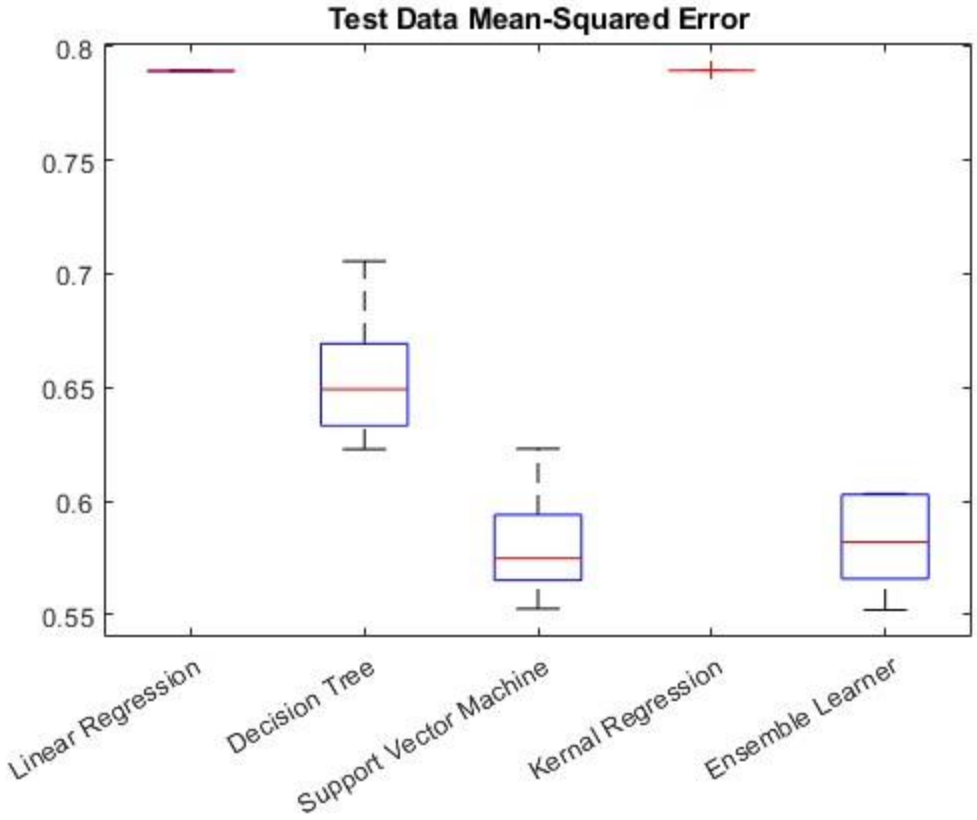

Figure 2 – Test-Data Mean Squared Error

Table 1 – Cross-Validation MSE

Lin. Regression	Decision Tree	SVM	Kernel	Ensemble
0.702770	0.703301	0.542102	0.743596	0.561407
0.748948	0.764075	0.725032	0.905634	0.594353
0.850765	0.657893	0.685186	0.840794	0.647760
0.865550	0.585976	0.650446	0.822734	0.593729
0.846307	0.699467	0.580353	0.780668	0.594845
0.732711	0.494683	0.549990	0.694095	0.605070
0.825300	0.698152	0.667366	0.774574	0.708120
0.687129	0.711628	0.615619	0.801305	0.537165
0.737956	0.649033	0.564992	0.794057	0.689584
0.862408	0.731460	0.688704	0.698193	0.606007

Table 2 – Test MSE

Lin. Regression	Decision Tree	SVM	Kernel	Ensemble
0.789060	0.669225	0.576534	0.789656	0.592940

0.789839	0.622771	0.585010	0.789648	0.570866
0.789605	0.664640	0.573111	0.789668	0.600864
0.789455	0.654832	0.623046	0.789622	0.552002
0.789424	0.705569	0.565062	0.789664	0.603243
0.789141	0.638072	0.558851	0.789653	0.558361
0.789373	0.699096	0.593956	0.789644	0.566860
0.789111	0.643573	0.552568	0.789562	0.602880
0.789296	0.633207	0.568275	0.789653	0.603255
0.789620	0.631888	0.615481	0.789688	0.565760

Table 3 – Model Average MSE

Error	Lin. Regression	Decision Tree	SVM	Kernel	Ensemble
CV-MSE	0.785984	0.669567	0.626979	0.785565	0.613804
Test-MSE	0.789393	0.656287	0.581189	0.789646	0.581703

As a result of the analysis completed on the result, the best performing model and thus the one to consider using in the future would be a support vector machine.

References

[1] The MathWorks, Inc. (n.d.). fitrlinear. Fit linear regression model to high-dimensional data - MATLAB. https://www.mathworks.com/help/stats/fitrlinear.html

[2] The MathWorks, Inc. (n.d.). fitrtree. Fit binary decision tree for regression - MATLAB. https://www.mathworks.com/help/stats/fitrtree.html

[3] The MathWorks, Inc. (n.d.). fitrsvm. Fit a support vector machine regression model - MATLAB. https://www.mathworks.com/help/stats/fitrsvm.html

[4] The MathWorks, Inc. (n.d.). fitrkernel. Fit Gaussian kernel regression model using random feature expansion - MATLAB. https://www.mathworks.com/help/stats/fitrkernel.html

[5] The MathWorkds, Inc. (n.d.). fitrenemble. Fit ensemble of learners for regression - MATLAB. https://www.mathworks.com/help/stats/fitrensemble.html