## **Computing Assignment 5**

## **Least - Squares Benchmark and 3 – Factor Wine Quality Predictor**

**Least – Squares Benchmark**

|  |  |  |
| --- | --- | --- |
| **Solve Type** | **Mean Coefficient Differences** | **RMS** |
| MATLAB’s backslash | N/A | 0.667349642444097 |
| Normal equation | 6.456e-08 | 0.667349642444098 |
| qr | 7.200e-13 | 0.667349642444098 |
| Permuted qr | 1.056e-13 | 0.667349642444097 |

*Table 1: Benchmarking results*

MATLAB’s backslash least-squares solve was benchmarked by comparing it to normal equation solves, and MATLAB’s ‘qr’ function with and without permutation. The results from table 1 reveal that MATLAB’s backslash might involve a permuted ‘qr’ function call, since the RMS is identical, and their coefficients are very similar.

**3 – Factor Wine Quality Predictor**

The goal of this model is to predict the quality of wine based on three given factors using supervised learning. The data used for this modal has ten factors (columns), labelled each being some numerical attribute for wine. Our model will use three of these factors, specifically the three which yield the smallest relative RMS residual from the training data (white.csv): j = {j1, j­2, j3}.

These optimal hyperparameters were found by evaluating the RMS for every possible combination of three factors, a technique called grid search. The smallest RMS found from grid search used:

j = {2, 4, 8} with RMS 0.7811,

which corresponds to the numerical attribute’s volatile acidity, residual sugar, and density. Next, this model must be trained to predict ratings for wines based on the three optimal factors. Using the method of least-squares yields the following equation:

,

where ,

and is the quality variable for each wine. Having [A] constructed with our optimal factors, and given in the training data, the least-squares coefficients vector can be solved for using MATLAB’s backslash, thus training our model. For any matrix [A] of wine data, the corresponding quality rating can be predicted using this trained vector .

|  |  |  |
| --- | --- | --- |
| **Predicted Quality** | **Actual Quality** | **Bottle Number** |
| 7.10 | 7 | 4285 |
| 7.05 | 7 | 2135 |
| 6.99 | 8 | 807 |
| 6.99 | 8 | 3625 |
| 6.94 | 6 | 1508 |

*Table 2: Quality Prediction Using Training Data*

As a test, the model was used to predict wine ratings from the training data; the results sorted in descending order by ratings is shown in table 2. These results are all within the expected accuracy range of ± RMS, validating our model. The highest rating in the training data is 8, whereas our model predicted it to be 7.1. From this it can be deducted that our model under predicts the highest wine quality.

|  |  |  |
| --- | --- | --- |
| **Rank** | **Predicted Quality** | **Bottle Number** |
| 1 | 6.77 | 14 |
| 2 | 6.74 | 25 |
| 3 | 6.69 | 40 |
| 4 | 6.55 | 35 |
| 5 | 6.53 | 42 |

*Table 3: Quality Prediction using whitelist2.csv Data*

The model was then used to predict wine ratings from the untrained data in whitelist2.csv. Table 3 shows the top five predicted wine ratings in whitelist2.csv.