It is widely acknowledged within the textiles industry that ratings-based data is ‘noisy’. While the models produced by this study were successful in identifying trends between yarn characteristics and user rating, any predictions made from these models are highly prone to error; and no single characteristic can be said to have an overriding influence on the rating. Indeed, the noisy nature of the yarn data implies a ceiling on the accuracy of any model, no matter how well trained it is.

However, the author acknowledges that improvements on the model accuracies presented here are certainly possible. To obtain improved results while keeping computation time/cost constant, the following may be investigated in the future:

*Improved computational efficiency*

In all cases, models were trained and tested using parallel code running on 36 virtual CPU cores. The first step to improve efficiency would be to optimise the parallelisation. For example, for the SVM models, a single thread could delay the entire training run by occupying a single CPU core for several hours. If the SVM algorithm itself was parallelised, rather than just the parameter tuning and cross validation, a speedup could be achieved.

However, a more effective approach would be to use GPU processing. This would allow several thousand threads to be run simultaneously, therefore sampling from a much larger parameter grid and evaluation of a greater number of candidate models. GPU processing was not used for this study as libraries for GPU-optimised code are only available for limited applications in R currently. The deepLearning function in H2O may be run on a GPU, however this functionality was not used in order to allow for a fair comparison between models.

While it is certainly possible to manually create GPU-optimised functions in R, a better approach currently would be to use a dedicated alternative package within an alternative language, such as TensorFlow or MXNet REFERENCE.

*Parameter optimisation*

During generation of candidate neural network models, it was apparent that the method of randomly searching the parameter space was highly inefficient. Rather than a random or grid search (as used for the SVM and Random Forest models), some form of optimisation algorithm could potentially be used to select the best parameters. A genetic algorithm or simulated annealing are possible candidates and could be used in conjunction with GPU processing to ensure enough candidate models were generated to make the optimisation worthwhile (Elyan and Gaber, 2016).

*Alternative models*

This study focused on three distinct algorithms. Variations of these methods, or different methods entirely may give improved results.

* In the case of SVMs, different kernel methods may be used to the radial basis function applied here
* Numerous varieties of neural networks are available in addition to the deep backpropogation network used in this study, which have been shown to exhibit excellent performance for certain problem types. REFERENCE
* Alternatives ensemble learning methods to the Random Forest model include Boosting, Bagging and Stacking. REFERENCE

Other, completely different, machine-learning methods are available which may out-perform the Random Forest. Unsupervised clustering methods, for example, may give further insight into which variables are the most important for determining yarn rating.

*Data processing*

The full dataset was not used for this study – approximately 40% was omitted due to NA values. This decision was made in order to simplify analysis and speed up computation time. Future studies may, however, benefit from assigning values to these NA entries in order to make use of the full dataset.

Similarly, a number of text-based attributes were removed from the dataset. These contained information such as a written description of the yarn, and information on the supplying company. Consideration for these text-attributes could provide improved results but would require alternative methods of analysis which were outside of the scope of this study.