Approximately 40% of the dataset entries contained NA values. Ideally, the NA values would have been substituted for replacement values REFERENCE. However, it was decided to omit these entries from analysis due to the already large dataset size and to simplify analysis. In addition XX instances had a yarn rating of zero. These instances represented yarns which had not yet been rated. These were also removed from the dataset as these instances formed a discontinuity in the raw data – zero star ratings are not possible on Ravelry so with these instances included, yarn ratings could take the values 1.0-5.0 OR 0.0. Removing the non-rated instances ensured the data was continuous and therefore suited to a regression model. A total of XX instances remained for analysis.

INSERT RATINGS DISTRIBUTION

INSERT PCR SECTION

Due to the inability of PCR to identify a clear set of important variables, it was decided to construct a self-organising map (SOM) to assist in visualising any potential relationships between variables. Through train and error, a grid size of 40X40, with a hexagonal topography was chosen to ensure instances were reasonably well distributed amongst nodes in the map. Separate heatplots for each individual variable were created from the output of the SOM algorithm. Each heatplot provided a visual representation of the distribution of its respective variable throughout the map. The positions of each instance from the dataset remained constant between heatmaps, which were coloured by the ‘density’ of the respective variables REFERENCE. Correlations between variables can be qualitatively identified by studying the heatplots for similar or differing regions across two plots. No immediately obvious correlations were identified but some more subtle potential relationships are demonstrated in figure XX – INSERT FIGURE.

The inability to identify any clear correlations in the SOM heatplots further suggests that no single variable, or simple linear combination of independent variables has a significant influence on the yarn rating. As a result, more complex methods of analysis were necessary to identify possible non-linear relationships.

All data acquisition, processing and analysis were performed using R. In order to reduce computation time, the data analysis was performed with virtual Linux machines using the Amazon EC2 cloud computing service. In all cases a single ‘4.8x Large’ instance (36 core Intel Xeon E5-2666 v3) REFERENCE was used, running RStudio server. Prior to using EC2, attempts were made to train models on subsets of the data (with 10% of the dataset used for training), however such models gave surprisingly inconsistent results; therefore the entire dataset was used for model training and testing going forward. In order to effectively utilise the multi-core architecture, parallelisation of the code was necessary. In the case of the SVM and random forest models, parallelisation was carried out on the parameter grid search and the cross validation process. For, example 5 fold cross validation for a single set of parameters was split into 5 threads which were run in parallel – each on a single processor core. In order to implement this, the ‘doParallel’ library was used REFERENCE. The deeplearning function in H2O, as used for the neural network, is implicitly parallel through the *Hogwild!* scheme (Niu et al, 2011) so no further parallelisation of the code was required. Each method was limited to approximately 4 hours of computation time to allow a meaningful comparison between the methods. In all cases, some experimentation was carried out on a local machine with small subsets of the dataset to determine rough ranges for tuning parameters prior to training the full dataset on EC2.

5 fold cross validation was used throughout to evaluate each model. This was chosen as a compromise between evaluation quality and computation time. As is common for regression models, the Root Mean Squared Error (RMSE) was used as the evaluation metric for each model. Prior to computing any predictive models, the RMSE between all actual yarn ratings and the mean yarn rating was calculated, giving a RMSE of 0.675. As this was a trivial computation, this was used as a benchmark for model performance, with any RMSE below this considered an improvement.