1 Introduction

We examined survey response data from the LISS panel survey in the Netherlands. Survey responders were asked whether they found the survey "interesting" at the end of the survey on a scale of 1 to 5. The surveys types were of 3 categories: leisure, health and income. Our goal was to predict the survey "interest" level based on various data from the LISS panel. Data included various background and socio-economic information of the responder, as well as time and dates when the survey was completed.
In order to predict the survey "interest" level, various classification techniques were employed. The metric used to analyse the performance of our technique was the multi-logarithmic loss (LL). Let $\mathtt{y_{ij}}$ be an indicator variable whether class j was correct for case i. Let $\mathtt{p_{ij}}$ be the probability of classifying into class j for case i. Our goal was to minimize this loss, which can be written by the following equation:

$$\text{Multi Log Loss} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \tag{1}$$

2 Data

The data was provided in two source files for training and testing, which needed to be merged on a key identifier. This resulted in missing data across various variables for matched rows. Further, standard feature engineering was required to ensure the data was optimal for use. Below is an overview of methods employed to prepare the data for modelling.
We used the `rfimpute`() function in the R randomForest package, to impute the N/A's in the training data. The method starts off using median for continuous predictors and mode for categorical variables. Then it iterates over the proximity matrix to update the imputation. For continuous variables, the imputed number is a weighted average of the observations, and for categorical variables, the imputed value is the factor with the largest average proximity. For the test set, we just use the median/mode method to fill in the N/A's.
For feature engineering, we chose to take the logarithm of all the income variables. If the income level was 0 then we took the log of 0.001 to avoid error.
Since the survey "interesting" rating was the response variable, and some information for previous interesting ratings was available, it was possible to add a historical interesting rating as an input variable. This was done by aggregating the median "interest" rating over each participant Id, for previous surveys completed, to assign a "historical" interest rating per partipcipant. For participants who had no previous survey submissions, the missing value was assumed to be the median "interest" rating over all responses available.

3 Preliminary Methods

Several methods are examined to find the best suited model for the data and required prediction. These methods include Linear and Quadratic Discriminant Analysis, Multinomial Regression, Random Forest, Naïve Bayes, and Support Vector Machines. The Support Vector Machine method resulted in the lowest log loss error for prediction, and so a more

detailed view into this method will be presented, in Section 4. All other methods will present overall issues, constraints, and high level results.

3.1 Linear Discriminant Analysis (LDA)

The first task taken on for fitting the LDA model was to find the variables that were most impactful to a good prediction. To accomplish this, each variable was individually fit to the response with an LDA model, and log loss prediction values were recorded. The log loss was ordered from least to greatest, with the corresponding variable associated to the loss. A model was then grown starting with the smallest log loss variable, and at each incremental step, the log loss prediction error was recorded, along with the variables utilized in the model fit.

After all potential variables were added to the model, the log los errors were compared to each other, and the model resulting with the smallest log loss prediction error was chosen. The resulting model included 28 variables; 26 factors and 2 continuous variables.

Generally, one would want to ensure the assumptions made for fitting the LDA model are satisfied, to ensure the model fit is appropriate for the data. This was not initially considered, since the goal was a low prediction error. After the model fitting was completed, the assumptions were reviewed. The resulting model includes categorical variables, for instance Marital Status and Housing, which are clearly not from a Normal Distribution. The variables that are continuous, for instance Duration, do not all follow a Normal Distribution. As such, the distribution assumption is not satisfied.

Since the variables do not meet the Normality assumption for LDA, they do not satisfy the assumption of equal variance/covariance. Two main assumptions of LDA are not satisfied with this dataset. Regardless, including these variables resulted in the lowest log loss for this model, and so were maintained within the model.

The inclusion of the historical "interest" rating improved the test prediction results. The log loss for prediction error improved by approximately 0.22. This proved to be a valuable variable to include, however is another categorical variable fit in the model.

3.2 Quadratic Discriminant Analysis (QDA)

Similarly to LDA, the first task taken on for fitting the QDA model is finding the variables that are the most impactful to a good prediction. The same approach is utilized for QDA, where each variable is first modelled individually via QDA against the response. Based on the ordered log loss prediction error, the full QDA model is grown variable by variable, and the lowest log loss error is determined. The historical interest rating was not one of these variables to remain in the model.

After fitting the best model for QDA, assumptions were reviewed. QDA holds the same assumptions as LDA, except for the equal variance/covariance. As detailed in LDA, the normality assumptions do not hold for the data for the QDA model. This is due to the categorical variables included and the continuous variables not satisfying normality tests. Again, the model fitted was kept, as it did result in the lowest log loss error for QDA.

3.3 Multinomial Regression (MR)

Multinomial Regression was not reviewed in class, however, is an extension of Logistic Regression that was presented during lectures. MR provides a method in which to perform classification using logistic regression for more than two classes, K=1,..., k. Essentially, the same principles are used, however, the resulting model requires additional calculations for each of the class probabilities. The probabilities are computed against a reference category as in the equations below, for reference category k, where r=1,..., k-1 (Teisseyre, P. 2014).

$$P(Y = r|x) = \frac{\exp(x^T\beta_r)}{1 + \sum_{s=1}^{k-1}\exp(x^T\beta_s)}, \quad P(Y = k|x) = \frac{1}{1 + \sum_{s=1}^{k-1}\exp(x^T\beta_s)} \tag{2}$$

The reference category is used in interpreting the resulting log-odds. Since the results are calculated against the reference category, there is a direct comparison to it. Figure 1 (Rodriguez, G. 2007) below shows such an interpretation for a model with 3 classes, in which the two plotted log-odds are in reference to the class of "No Method" for that example.

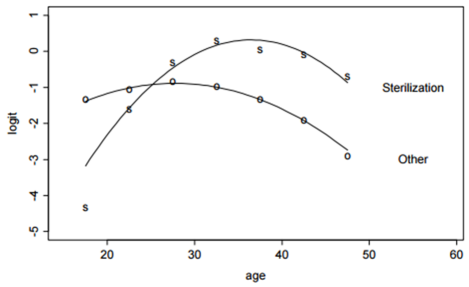

Figure 1: Interpretation of 3 class Multinomial Regression model (Rodriguez, G. 2007).

The same approach was utilized for fitting the MR model as in LDA and QDA. Including the historical "interest" ratings, the log loss prediction error improved by ∼0.17. The training data resulted in a log loss value of ∼0.606 using only 5 variables. This model showed much improvement over the DA models.

3.4 Naïve Bayes (NB)

Naïve Bayes assumes the distributions of all predictors are independent. Features were selected through the "Recursive Feature Elimination" (RFE) technique found in the caret package in R. "Accuracy" was used as a model metric. The algorithm is as follows:

1. Train model using all predictors.
2. Calculate the model performance using accuracy.
3. Calculate variable importance by the area under the curve (AUC) metric.

   For all variable subsets Si:

   a) Keep the top Si variables;

   b) Train model using the Si variables;

   c) Calculate model performance using accuracy

4. Calculate performance over all Si.

5. Determine the right model based on the Si with the best accuracy.

Using RFE with Laplace smoothing n=1 and 5-fold cross validation, 34 variables were fit. The model was trained on 100% of the training data, which resulted in a MLL of 1.503. RFE was used on the feature engineered data with the same Laplace smoothing and cross validation. The algorithm included birth year as an optimal variable, with a resulting model accuracy of 0.3698. The model was trained using birth year only, and resulted in a MLL value 1.396 on the training data. Overall, this method was not suitable for Kaggle submission.

3.5 Random Forest (RF)

All the random forest experiments were run using the randomforest R package. Experiments were run with 300 trees, since the OOB error rate was lowest for 289 trees when the model was run with all variables, 5-fold cross validation and "mtry=10" at each node.

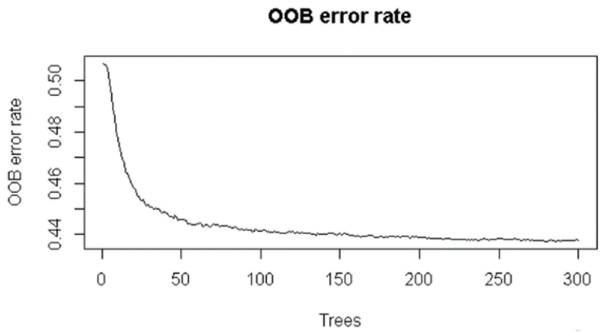

Figure 2: OOB error rate as a function of the number of trees

Three approaches were tried for RF. First, a model was run with all the variables without any sort of feature engineering. This gave a Kaggle MLL of 1.511. Second, a two-model approach was used: one model for data with all background information available and a second model for the data without background information. Just *type* and *duration* variables were included in the second model, and the resulting two-model gave a Kaggle MLL of 1.507. Last, a model was fit with the feature engineered variables on all predictors, except *brutoink*, *nettoink*, *netinc*, *brutocat*, *nettocat*, *brutohh_f*, *nettohh_f*. This resulted in a Kaggle MLL of 1.273.

4 Final Method: Support Vector Machines

The modeling of the survey interest data by support vector machines (SVM) produced the lowest log-loss number of all the methods implemented. Since this is the best method over all the other implementations, this method is described here in detail.
The support vector machine method is a classification algorithm based on partitioning the feature space by a hyper-plane in $\Re^n$. The general equation of an n-dimensional hyper plane is given by: $f(x) = \beta^T x + \beta_o = 0$, where β is an n-dimensional vector of constant coefficients. The set of training observations are denoted as $\{x_1, y_1\}, \{x_2, y_2\}, ..., \{x_N, y_N\}$

where the response variable is binary and assigned $y_i = -1 or 1$. In the case of a feature space which is linearly separable by training class, the SVM method finds the unique β and β_o which solve the optimization problem:

$$\max_{\beta,\beta_o,\|\beta\|=1} M, \text{ s.t. } \quad y_i(\beta^T x_i + \beta_o) \geq M, i = 1, 2, ..., N \tag{3}$$

Here M is the distance of $f(x) = 0$ from a parallel plane on either side which intersects the closest training point in the feature space. In other words, this is the margin on either side of $f(x) = 0$ which contains no training observations. Intuitively, finding such a maximal M ensures that a small amount of random noise in an observation does not result in a change in classification. To justify the condition $\|\beta\| = 1$, note that for any two points x_1 and x_2 lying on the hyper-plane H given by $f(x) = 0$:

$$f(x_1) - f(x_2) = (\beta^T x_1 + \beta_o) - (\beta^T x_2 + \beta_o) = \beta^T (x_1 - x_2) = 0.$$

Hence $\beta/\|\beta\|$ is a unit normal to H. Let $x_o \in H$, so the signed perpendicular distance of any point x to H is:

$$\frac{\beta^T}{\|\beta\|}(x - x_o) = \frac{1}{\|\beta\|}(\beta^T x - \beta^T x_o) = \frac{1}{\|\beta\|}(\beta^T x + \beta_o) = \frac{1}{\|\beta\|} f(x).$$

So setting $\|\beta\| = 1$ implies $f(x)$ is the signed perpendicular distance of a point x to H.
The assumption of linear separability can be relaxed by introducing slack variables ϵ_i such that $\epsilon_i \geq 0$. These are defined to be the proportion by which training observation i is on the wrong side of the margin. The constraint is restated as:

$$y_i(\beta^T x_i + \beta_o) \geq M(1 - \epsilon_i).$$

As an example, if x_i is on the correct side of $f(x) = 0$ but in the middle of the margin, then $M(1 - 1/2) = M/2$, and the restriction is lowered for this observation. The form of the optimization problem given in (3) allows for an immediate intuitive interpretation, but is not convenient for computation purposes. An alternative formulation would be to instead choose $\|\beta\| = 1/M$, and so maximizing M is equivalent to minimizing $\frac{1}{2}\|\beta\|^2$. Putting all this together, the optimization problem is restated as:

$$\min_{\beta,\beta_o} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \epsilon_i, \text{ s.t. } \quad \epsilon_i \geq 0, \quad y_i(\beta^T x_i + \beta_o) \geq 1 - \epsilon_i, i = 1, 2, ..., N. \tag{4}$$

Here $C \in \Re$ is a scaling factor called the cost. If observation i is inside the margin by ϵ_i, then the penalty for decreasing the margin is to increase the objective function by $C\epsilon_i$. Hence a larger C places a higher penalty on points on the wrong side of their margin. The final aspect to consider for SVM is the selection of the kernel. To illustrate this concept, consider the Lagrangian of the optimization problem (4):

$$L = \tfrac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N} \epsilon_i - \sum_{i=1}^{N} \lambda_i(y_i(\beta^T x_i + \beta_o) - (1 - \epsilon_i)) - \sum_{i=1}^{N} \zeta_i \epsilon_i$$

Here, λ_i and ζ_i are the general Lagrange multipliers. The derivatives are taken with respect to β, β_o, and ϵ_i, but we state only the derivative w.r.t. β:

$$\beta = \sum_{i=1}^{N} \lambda_i y_i x_i \quad (5)$$

In order to better separate the feature space, we may choose to apply an appropriate transformation of the form: $x \mapsto h(x)$. Combining this with (5), the estimate $\hat{f}(x)$ can be stated as:

$$\hat{f}(x) = h(x)^T \hat{\beta} + \hat{\beta}_o = \sum_{i=1}^{N} \hat{\beta} \lambda_i y_i h(x)^T h(x_i) + \hat{\beta}_o.$$

Note that most of the λ_i are zero except for the ones corresponding to the support vectors, of which there are usually few. Hence we predict $y = 1$ if $\hat{f}(x) > 0$ and -1 otherwise. So the classification depends only on the inner product of the support vectors and we do not need to know $h(\cdot)$ explicitly. The kernel, defined as the inner product below, has a specific form dependent on the problem being considered:

$$K(x, x') = h(x)^T h(x')$$

4.1 SVM Applied To Survey Data

The application of the SVM method to the survey data set was done using the e1071 package [4] in R. This package extends the SVM method described above to the case of a multiclass classifier through the 'one-against-one' method. For our 5 classes, this entails training 5(5-1)/2=10 classifiers for each binary pair of "interest" categories $\{1, 2, 3, 4, 5\}$, and deciding on the final class by majority vote. The package scales the data to be applied to the SVM method. The choice of kernel function was done by testing a range of tuning parameters on 1/10th of the data on the linear, polynomial, and radial kernels and choosing the kernel with the lowest Log-Loss. The values of C were tested from $\{2^{-5}, 2^{-3}, ..., 2^9\}$. For the polynomial kernel the degrees were tested from the set $\{2, 3, 4, 5\}$. For the radial kernel given by:

$$K(x, x') = exp(-\gamma \|x - x'\|^2),$$

the γ values are taken from $\{2^{-14}, 2^{-12}, ..., 2^2\}$. The grid-search method with ranges of C and γ defined above are suggested in [3]. The radial kernel proved to be the best performing of the three methods. Widely varying C was found to not significantly change our results, suggesting that the data is well separated in the feature space. Table 1 provides a subset of the $C \times \gamma$ grid where our values were taken from, where G is γ.
Note that these are the Log-loss values obtained on the 75-25 split of the train-test data. The chosen parameters were $C = 0.125$ and $\gamma = 0.003906$, since the LL above and below are about the same and hence varying C and γ slightly does not change our LL significantly. The best LL obtained on the Kaggle set was 1.153 with these parameters, training on the entire data set.

C	G	LL
0.5	6.10E-05	0.341657
2	0.000122	0.345721
8	0.000244	0.342092
32	0.000488	0.341668
128	0.000977	0.337762
512	0.001953	0.335817
0.03125	0.003906	0.345569
0.125	0.007813	0.348443
0.5	0.015625	0.345567
2	0.03125	0.361976

Table 1: Training Set Log-Loss

5 Conclusion

Careful data processing was required prior to beginning any model fitting for this dataset. Feature engineering and variable manipulation increased the predictive power in all models fitted. The log loss for most models fit, for training and test date, returned values below 1, some below 0.5. However, these results did not materialize when fitting the models to the final test data submitted in Kaggle. For instance, the best QDA model resulted in a training/testing log loss of ∼0.606 vs a Kaggle log loss of 1.357. This was a common result through our models fit and tested. As of the submission date of this paper, the Support Vector Machine model has the lowest Log-Loss of 1.15346.

6 References

[1] P. Teisseyre, 2014, Lab: Multinomial model, Institute of Computer Science, http://www.ipipan.eu/∼teisseyrep/TEACHING/SML/Multinomial%20model.pdf

[2] G. Rodrguez, 2007, Lecture Notes on Generalized Linear Models, http://data.princeton.edu/wws509/notes/

[3] C. Hsu, et al., 2016, A Practical Guide to Support Vector Calssification http://www.csie.ntu.edu.tw/∼cjlin/papers/guide/guide.pdf

[4] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch (2015). e1071:Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6-7. http://CRAN.R-project.org/package=e1071

[5] M. Kuhn. Contributions from J. Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem, L. Scrucca, Y. Tang, C. Candan, 2016. caret: Classification and Regression Training. R Package version 6.0-71. http://CRAN.R-project.org/package=caret

[6] A. Liaw, M. Wiener, 2002. Classification and Regression by randomForest. R News 2(3), 18–22.