Prediction of Supermarket Turnover using Demographic Data

Group 3: Caio Rego, Irene Vega Ramón, Jan Janiszewski, Sjoerd Stevens

08/11/2021

1. Introduction

To predict where a new supermarket should be built is essential to supermarkt chains and the competitive edge gained from good predictions is integral to the success of a supermarket chain. However, there are many predictors of supermarket turnover as the shopping behavior of supermarket customers is an interaction of different factors which are not clearly found at first sight.

Therefore, this paper aims to analyze if supermarket turnover can be predicted using a subset of demographic data. Based on this goal, we have two research questions, which are: - Can supermarket turnover be predicted using demographic data? - Which subset of demographic variables minimizes our prediction error for predicting supermarket turnover?

2. Data

Data used to answer our introduced questions originates from the U.S. government census (1990) and describes the demographic data for different supermarkets in the Chicago metropolitan area. It is part of a bigger data collection of data on 77 supermarkets in the Chicago area, which has been used for research on shelf management and pricing (Kilts Center, 2004).

The data can be grouped into three categories, namely variables describing the supermarket in a given area, demographic characteristics of population living in the surroundings of the supermarket, and their shopping behavior (most likely from questionnaire data). It consists of 77 rows representing the 77 supermarket that have been researched (with one row covering the area of one supermarket). Each row consists of 44 different float values representing different customer or store characteristics. Using those variables, we want to predict total turnover in one year of groceries in $ for a given supermarket (grocery_sum; $Min = 1423582\$, Mean = 7341015\$, Max = 13165586\$$)

For further information on the variables and the descriptions, see Appendix 1.

As our research question is not related to location itself but rather to demographics accompanying the location and shopping behavior, we are not using variables related to the store, city, and zip code. Also, as grocery coupon turnover is only indirectly related to our question of the total turnover of a supermarket (it affects it negatively), we are also not using this variable in our data.

Due to the high quality of the dataset we received (no bogus answers, no incorrectly values, no extreme or outlier values), no further preparation steps to the data were necessary.

For in-depth descriptions of the data (e.g. mean), please see Kilts Center, 2004.

3. Methods

A multiple regression model is used to predict the total supermarket turnover given the set of demographic data, which assumes that the regression function $E(Y|X)$ is linear in the inputs $X_1, ..., X_p$. First, to reduce variance at the cost of bias, a shrinkage method is applied to the set of predictors. Specifically, the elastic net is chosen for its ability to select variables by soft thresholding like the Lasso and to apply proportional shrinkage to such variables like the Ridge regression. Second, the adjusted set of predictors is analyzed using the linear regression model that includes the penalty for large predictor set. Third, the expected prediction error is estimated using K-fold cross validation.

First, the standard linear regression model is:

$$y = X\beta + \epsilon,$$

where $X = (x_{ij}, ..., x_{np})$ is the n × p predictor matrix, $y = (y_1, ..., y_n)$ is the predicted vector,$\beta = (\beta_1, ..., \beta_p)$ is the vector of regression coefficients corresponding to $X_1, ..., X_p$, and $\epsilon = (\epsilon_1, ..., \epsilon_n)$ is a vector of i.i.d. random errors with $\overline{\epsilon} = 0$ and $Var(\epsilon) = \sigma^2$. X is scaled by standardizing each X_{ij} to be able to compare all predictor variables and by transforming the intercept on y to $\overline{y}$.

The loss function used to apply the shrinkage of the predictor matrix is:

$$L(B) = (y - X\beta)^T(y - X\beta) + \lambda \sum_{j=1}^{p} (\alpha\beta_j^2) + (1 - \alpha)|\beta_j|),$$

where the addition to the sum of square residuals is a penalty for the size of the predictors, thus aiming for more variance explanation at the cost of biasedness. Specifically, the penalty factor is constructed as follows. λ is a complexity parameter that controls the amount of shrinkage, with larger λ being greater shrinkage. α is a mixed parameter between the Ridge regression and the Lasso. This allows the penalty to capture the flexibility of the Ridge with the variable selection of the Lasso. First, β_j^2 corresponds to the Ridge regression parameter, a quadratic penalty that makes the Ridge regression a linear function of y. The solution adds λ to the diagonal of X^TX before inversion, thus making the problem non-singular even if X^TX is not full rank. By doing this, coordinates are shrinked by a factor $\frac{dj^2}{dj^2+\lambda}$, being dj small singular values that correspond to the directions where the column space of X has small variance. Second, $|\beta_j|$ corresponds to the Lasso penalty, a type of continuous subset selection that causes the weakest predictors to be set to 0 if λ is sufficiently large.

After the shrinkage method on the prediction matrix has been applied, the effect of the prediction matrix on the predicted vector is analyzed using the coefficient matrix. The diagnostics used for evaluating fitness are the magnitude of the Root Mean Square Error (RMSE) and the adjusted R^2, being smaller RMSE and larger R^2 account for stronger predictors.

Expected prediction error is estimated using the K-fold cross-validation by applying the shrinkage method and multiple regression ($\hat{f}(x)$) to an independent test sample. Specifically, data is split into k equal parts, The k-th part is used to calculate the prediction error of the fitted model, using the objective function:

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i)).$$

In this report, a choice of $k = 10$ is made such that cross-validation has lower variance, accepting that bias might be higher than a choice with a lower k. The parameters λ and α are determined after the cross-validation, using those that minimize the RMSE.

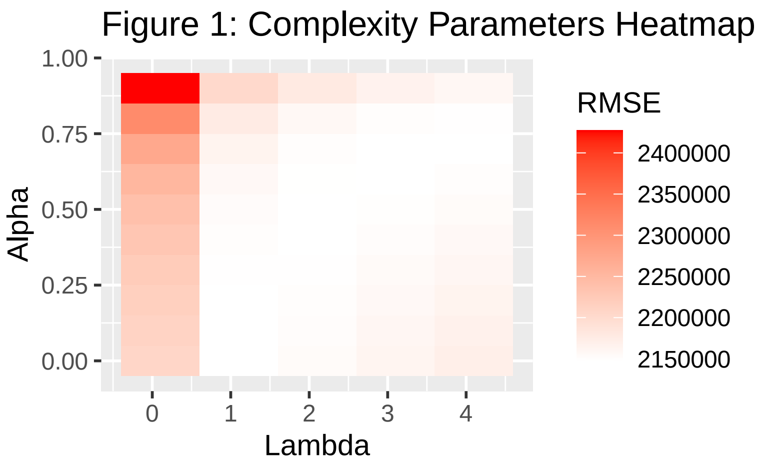
4. Results

The K-fold cross validation yields an $\alpha = 0.5$ and $\lambda = 0.1$ such that the minimum RSME is obtained, as illustrated in Figure 1. This indicates that the best results are achieved by constructing the penalty using equally the Lasso and the Ridge parameters, and by applying a relatively large penalty to the least square error estimate.

Table 1

MM Elastic Net and GLMNET Elastic Net Model Diagnostics

	R^2	RMSE
MM	1782121	0.4593
GLMNET	3263000	0.367



Using these complexity parameters, the model yields a RMSE = 1782121 and the $R^2 = 0.4593$, as observed in Table 1. Although there is a relatively high RMSE, this is partially due to the large values of y. The predicted values are capable of accounting for approximately 45.9% of the variance of the observed values for y.

The prediction is significantly better than the one generated by the GLMNET standard package, with an R^2 value of 0.459. The difference in computation is due to GLMNET's (1) least square error term division by $2n$ and (2) division of the $(1-\alpha)$ parameter (corresponding to the Ridge regression) by 2, thus making half of the shrinkage performed by our model. It is unclear to us why GLMNET divides the least square error term by 2 if it does not do so in neither the Lasso or the Ridge regression.

5. Conclusions and Discussion

The aim of this research was to predict supermarket turnover using demographic data. The research adjusted the predictor matrix using the elastic net, which eliminated in a continuous way the weakest predictors and adjusted their coefficients. The adjusted set of demographic variables account for approximately 46% of the variance of supermarket turnover in our model. This is highly relevant for the supermarket industry, for it can significantly predict which supermarkets will have a higher turnover or create supermarkets in the most favourable demographic circumstances, thus improving the conducting of operations and the collection of cash from accounts receivable.

The research did not focus on the nature of the correlation. To manipulate the increase in supermarket turnover, future research can investigate if there is a causal relation between these variables through different methods such as experiments.

6. Appendix

6.1. Variable Description

6.2. Code

```
# RSS: Calculates the residual sums of squares
#
# Parameters:
#     X: matrix, contains independent variables
#     y: vector, contains the dependent variable
#     B: vector, contains the coefficients
# Returns:
#     rss: a float representing the residual sum of squares
setwd("~/AA BDS/1-2 Supervised Machine Learning/Week 2")

RSS <- function(X,B,y) {
  y_hat <- X %*% B
  rss <- t(y - y_hat) %*% (y - y_hat)

  return(rss)
}

# RMSE: Calculates the root mean squared error
#
# Parameters:
#     X: matrix, contains independent variables
#     y: vector, contains the dependent variable
#     B: vector, contains the coefficients
# Returns:
#     rmse: a float representing the root mean squared error

RMSE <- function(X, Bk, y){
  rmse <- sqrt(RSS(X, Bk, y)/dim(X)[1])
  return(rmse)
}


# rsquared: Calculates the coefficient of determination
#
# Parameters:
#     X: matrix, contains independent variables
#     y: vector, contains the dependent variable
#     B: vector, contains the coefficients
#     adjusted: boolean, whether to calculate the adjusted R2
# Returns:
#     R2: a float representing the coefficient of determination

rsquared <- function(X, Bk, y, adjusted = FALSE){
  y_mean <- mean(y)
  SST <- t(y-y_mean)%*%(y-y_mean)
  n <- dim(X)[1]
  p <- dim(X)[2]
  R2 <- 1 - (RSS(X, Bk, y) / SST)
  if(adjusted == FALSE){
    R2 <- R2
  } else {
    R2 <- 1 - ((1 - R2)*(n-1))/(n - p -1)
  }
  return(round(R2,4))
}

# gridkcv: Generates the parameter grid to be used in a gridsearch
#
# Parameters:
#     start_lambda: float, smallest lambda to be part of the grid
#     end_lambda: float, biggest lambda to be part of the grid
#     delta_lambda: float, increment from lambda to lambda
#
#     start_alpha: float, smallest alpha to be part of the grid
#     end_alpha: float, biggest alpha to be part of the grid
#     delta_alpha: float, increment from alpha to alpha
#
# Returns:
#     grid: a matrix with all of the parameter combinations

gridkcv <- function(start_lambda, end_lambda, delta_lambda, start_alpha, end_alpha, delta_alpha){
  grid <- expand.grid(lambda = seq(from = start_lambda, by = delta_lambda, to = end_lambda),
                      alpha = seq(from = start_alpha, by = delta_alpha, to = end_alpha))
  return(grid)
}


# kfold: Generates the k-fold estimates (MSE, and R-squared)
#
# Parameters:
#     k: float, number of folds
#     X: matrix, contains independent variables
#     y: vector, contains the dependent variable
#     param_grid: matrix, the hyperparameter grid with the values to the be evaluated
#     verbose: boolean, wheter to return the loss function value at each iteration
#
# Returns:
#     fit_stat: matrix, for each combination in the parameter grid, it returns the fit statistics
#               mean-squared error and r-squared
#

kfold <- function(k, y, X, param_grid, verbose = TRUE){
  indices <- sample(dim(X)[1])
  folds <- split(indices, ceiling(seq_along(indices)/(length(indices)/k)))

  grid_size <- seq(dim(param_grid)[1])

  fit_stat <- param_grid

  for(j in grid_size){
    temp <- matrix(1, grid_size, 1)
    temp2 <- matrix(1, grid_size, 1)
    for(i in 1:k){

      y_train <- y[-folds[[i]]]
      X_train <- X[-folds[[i]],]

      y_test <- y[folds[[i]]]
      X_test <- X[folds[[i]],]

      fit <- elastic_net(y_train, X_train, param_grid[[1]][j], param_grid[[2]][j], verbose = verbose)

      coef <- fit[[1]]
      R2 <- rsquared(adj.designmatrix(X_test, scale = TRUE, intercept = TRUE), fit[[1]], y_test, adjuste
      rmse <- RMSE(adj.designmatrix(X_test, scale = TRUE, intercept = TRUE), coef, y_test)
      #print(param_grid[[2]][i])
      temp[[i]] <- rmse
      temp2[[i]] <- R2
    }

    fit_stat[j,'RMSE'] <- mean(temp)
    fit_stat[j,'R2'] <- mean(temp2)
  }
  return(fit_stat)
}

# adj.designmatrix : Adjusts the design matrix to accommodate an intercept if required
#
# Parameters:
#     X: matrix, contains independent variables
#     scale: Boolean, whether to scale the data
#     intercept: Boolean, whether to add a vector of ones (to support the intercept)
#
# Returns:
#     X: matrix, the adjusted design matrix
#

adj.designmatrix <- function(X, scale = TRUE, intercept = TRUE){
  if(scale == TRUE) {X <- as.matrix(scale(X))}
  if(intercept == TRUE) {X <- cbind(matrix(1, dim(X)[1], 1), X)}
}


# MJF : loss function to be used in each iteration of the MM algorithm
#
# Parameters:
#     B: vector, candidate coefficients
#     A: matrix, intermediate calculation used to update the coefficients
#     D: matrix, intermediate calculation used to update the A matrix
#
# Returns:
#     Bu: float, the loss function for that particular set of regressors
#
```

```
MJF <- function(B, A, D, xtx = xtx, xty = xty, yty = yty, lambda = lambda, alpha = alpha,n=n){
  c <- 1/(2*n)*yty + (1/2)*lambda*alpha*sum(abs(B))
  B_u <- 1/2*t(B) %*% A %*% B - 1/n *t(B)%*%xty + c

  return(B_u)
}

# elastic_net : generates the elastic net estimates for the regression, through the MM
#                algorithm
#
# Parameters:
#     X: matrix, contains independent variables
#     y: vector, contains the dependent variable
#     lambda: float, lambda
#     alpha: float, lambda
#
# Returns:
#     Bk: vector, the loss function for that particular set of regressors
#     R2:
#     RMSE:
#     X:
#     iterations:
#     loss:
#

elastic_net <-function(y, X, lambda, alpha, verbose = FALSE){

  X <- adj.designmatrix(X, scale = TRUE, intercept = TRUE)

  #pre-calculations
  p <- dim(X)[2]
  n <- dim(X)[1]
  xtx <- t(X)%*%X
  xty <- t(X)%*%y
  yty <- t(y)%*%y

  #set initial coefficient vectors
  B0 <- matrix(1, p, 1)
  Bk <- matrix(1, p, 1)
  epsilon <- 1e-6

  k <- 1
  L_b0 <- 10
  L_bk <- 9

  while (k == 1 | (L_b0 - L_bk)/ L_b0 > 1e-6){
    k <- k + 1
    B0 <- Bk
    inter <- apply(Bk, 1, function(x) max(abs(x),epsilon))
    I <- diag(p)
    I[1,1] <- 0
    D <- I/inter
    A <- (1/n)*xtx + (lambda * (1-alpha)) * I + (lambda * alpha) * D
    L_b0 = MJF(B0, A, D, xtx = xtx, xty = xty, yty = yty, lambda = lambda, alpha = alpha, n=n)
    Bk <- 1/n * solve(A) %*% xty

    L_bk = MJF(Bk, A, D, xtx = xtx, xty = xty, yty = yty, lambda = lambda, alpha = alpha, n=n)

    if(verbose == TRUE){
    print(L_bk)
    print(L_b0)
    }

  }
  R2 <- rsquared(X, Bk, y, adjusted = FALSE)
  rmse <- RMSE(X, Bk, y)
  res_list <- list('coefficients'=Bk,
                   'R2'=R2,
                   'RMSE'=rmse,
                   'iterations'= k,
                   'loss' = L_bk)
  return(res_list)
}
```

References

Hastie, T., Tibshirani, R. and J. Friedman. (2009). 'The elements of statistical learning (2nd edition).' Springer. Available at https://web.stanford.edu/~hastie/Papers/ESLII.pdf. J. M. Kilts Center. Dominick'sdataset. (2004). Available at https://www.chicagobooth.edu/research/kilts/datasets/dominicks. ***

Variable Name	Description
store	Store identification number
city	City of supermarket
Zip	Zip-code
grocery__sum	Total turnover in one year of groceries (in $)
groccoup__sum	Total of redeemed grocery coupons (in $)
age9	% Population under age 9
agebo	% Population over age 60
ethnic	% Blacks &. Hispanics
educ	% College Graduates
nocar	% With No Vehicles
income	Log of Median Income
incsigme	Std dev of Income Distribution (Approximated)
hsizeavg	Average Household Size
hsize1	% of households with 1 person
hsize2	% of households with 2 persons
hsize34	% of households with 3 or 4 persons
hsize567	% of households with 5 or more persons
hh3plus	% of households with 3 or more persons
hh4plus	% of households with 4 or more persons
hhsingle	% Detached Houses
hhlarge	% of households with 5 or more persons
workwom	% Working Women with full-time jobs
sinhouse	% of households with 1 person
density	Trading Area in Sq Miles per Capita
hval150	% of Households with Value over $160,000
hval200	% of Households with Value over $200,000
hvalmesn	Mean Household Value (Approximated)
single	% of Singles
retired	% of Retired
unemp	% of Unemployed
wrkch5	% of working women with children under 5
wrkch 17	% of working women with children 6-17
nwrkch5	% of non-working women with children under 5
nwrkch17	% of non-working women with children 6-17
wrkch	% of working women with children
nwrkch	% of non-working women with children
wrkwch	% of working women with children under 5
wrkwnch	% of working women with no children
telephn	% of households with telephones
mortgage	% of households with mortgages
nwhite	% of population that is non-white
poverty	% of population with income under $15,000
shopcons	% of Constrained Shoppers
shophurr	% of Hurried Shoppers
shopavid	% of Avid Shoppers
shopstr	% of Shopping Stranges
shopunft	% of Unfettered Shoppers
shopbird	% of Shopper Birds
shopindx	Ability to Shop (Car and Single Family House)
shpindx	Ability to Shop (Car and Single Family House)

Table 1: Variable description of variable used in the dataset