**ISYE 6502 HW1**

# Group members: Alan Lo, Arjun Goyal, Richard Hardis, James Trawick

**Question 2.1**

Describe a situation or problem from your job, everyday life, current events, etc., for which a classification model would be appropriate. List some (up to 5) predictors that you might use.

# Classification is useful in many aspects of professional and personal life. At The Home Depot,

# many different models and heuristics are utilized in order to make effective decisions.

# Specifically, building a classification model would be useful when making “a-bring-to-market”

# decision.

# In Merchandising/Product-Engineering, the age-old question is whether or not to bring a product

# to market. Product profitability is at the crux of this decision – as such, classifying whether or not a product will be profitable would greatly benefit The Home Depot decision makers.

# Although a probabilistic model might be a better reference when deciding how profitable one new item might be, a classification model is more useful when deciding whether or not to introduce one new item in a Boolean context.

# There are many predictors that would greatly improve a product profitability classification model – a good starting list of predictors might include: department, seasonality, raw material cost, supply-chain cost, and price.

Department is a categorical predictor. Including all of the Department (merchandise sectors) numbers:D21 – Lumber, D22 - Building Materials, D23 – Flooring, D24 – Paint, D25 – Hardware, D26 – Plumbing, D27 – Electrical, D28 – Seasonal, D29 - Kitchen and Bath, D30 – Millwork, D59 – Décor.

**Seasonality** is a categorical predictor. Including all four seasons: Winter, Spring, Summer, Fall.

**Raw material cost** is a continuous variable, taking on the cost to acquire the material used in product creation.

**Supply-Chain cost** is a continuous variable, taking on the ABC-based costs to travel upstream in the supply-chain.

**Price** is a continuous variable, taking on the price set by the marketplace/The Home Depot marketing decision makers.

Regarding the validity of these predictors, The Home Depot would require empirical validation. However, our team believes, a priori, these to be strongly correlated with profitability. In sum, a classification model would be a useful reference in “a-bring-to-market” decision.

**Question 2.2**

The files credit\_card\_data.txt (without headers) and credit\_card\_data-headers.txt (with headers) contain a dataset with 654 data points, 6 continuous and 4 binary predictor variables. It has anonymized credit card applications with a binary response variable (last column) indicating if the application was positive or negative. The dataset is the “Credit Approval Data Set” from the UCI Machine Learning Repository (<https://archive.ics.uci.edu/ml/datasets/Credit+Approval>) without the categorical variables and without data points that have missing values.

1. Using the support vector machine function ksvm contained in the R package kernlab, find a good classifier for this data. Show the equation of your classifier, and how well it classifies the data points in the full data set. (Don’t worry about test/validation data yet; we’ll cover that topic soon.)
2. You are welcome, but not required, to try other (nonlinear) kernels as well; we’re not covering them in this course, but they can sometimes be useful and might provide better predictions than vanilladot.
   1. Answer for Question 2.1 1-2:

We found that a good classifier for this dataset had the following equation:

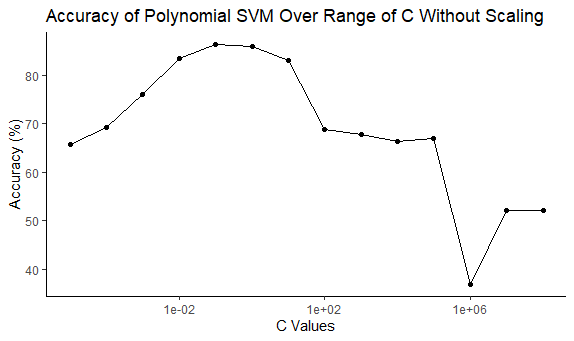
It was observed that the coefficient for predictor A9 was much larger than the coefficients for all other predictor variables. The accuracy of the selected model was **86.4%**. The nonlinear kernel “polydot” did not significantly impact the maximum accuracy of the model.

# A screenshot of a social media post Description automatically generated

A close up of a map

Description automatically generated

# A close up of a map Description automatically generated



1. Using the k-nearest-neighbors classification function kknn contained in the R kknn package, suggest a good value of k, and show how well it classifies that data points in the full data set. Don’t forget to scale the data (scale=TRUE in kknn).

Based on our exploration of the effect of K-values on the accuracy of the KKNN algorithm applied to the credit\_card\_data dataset, we found that the best value of K was **6**, with an accuracy of 85.2%. The accuracy was measured by number of correct classifications divided by the total number of classifications.

# R code output for KKNN Function

We write the function *kknn\_accuracy* to evaluate the k-nearest neighbors model using different values of k.

kknn\_accuracy = function(k\_val){  
 pred\_responses <- rep(0, nrow(cred\_card\_data\_headers)) #vector of 0s to insert the predicted response value of the model.  
 for (i in 1:nrow(cred\_card\_data\_headers)){  
 #using [-i] to exclude the ith data point from the nearest neighbor calculation  
 kknn.model = kknn(R1 ~., train = cred\_card\_data\_headers[-i,], test = cred\_card\_data\_headers[i,], k = k\_val, kernel = "rectangular",scale = TRUE)  
 #fitted(kknn.model) gives the value of the fraction of nearest neighbors to the ith data point that are 1.  
 fit <- fitted(kknn.model)  
 binary\_response <- as.integer(fit + 0.5) #maps the fitted value to 0 or 1  
 pred\_responses[i] <- binary\_response #adds the model's fitted value to the pred\_responses vector  
 }  
 acc = sum(pred\_responses == cred\_card\_data\_headers[,11]) / nrow(cred\_card\_data\_headers)  
 return(acc)  
}

We run a for loop iterating through [0,20] for values of k.

k\_acc <- rep(0, 20)  
for (i in 1:20){ #values of k  
 k\_acc[i] = kknn\_accuracy(i)  
}

To determine the best accuracy delivered by the model, we run:

max(k\_acc) #best k accuracy

## [1] 0.851682

To determine the value of k for which this best accuracy occurs, we run the following code to find the index of the maximum accuracy:

max\_k <- which(k\_acc %in% max(k\_acc))  
max\_k

## [1] 6

#Plotting Accuracies

We can create a plot of the values of k vs. the accuracies: 