# TIM245: Data Mining

**Homework #2:** Due Wednesday, 10 May 2017

**Instructions for Homework # 2:**

* You are allowed to discuss homework problems with other members of course, however, your problem solutions must be **distinctly your own work**, and not a copy of any other student’s work.
* In the assignment you will make use of the following tools: RStudio, and Weka. Before starting the assignment, please download the tools from the links provided below:

1. RStudio ([https://www.rstudio.com/products/rstudio/)](http://openrefine.org/))
2. Weka (<http://www.cs.waikato.ac.nz/ml/weka/>)

* Please organize your answers to the bolded questions into a well-structured 4-6 page report and submit a hard-copy in class on Monday.

**Problem Statement**

In this homework assignment, you will create a supervised learning model for two important problems related to peoples’ happiness:

1. Predicting the overall happiness of the citizens for different countries around the world.
2. Predicting customer churn at a large telecommunication company.

Before starting the assignment, please download the World Happiness and Telcommunications churn datasets from the course webpage: <https://tim245-spring17-01.courses.soe.ucsc.edu/>

**Problem 1: Linear Regression Modelling of World Happiness Scores**

*Note: before starting this assignment it may be useful to review the tutorial* [*http://web.cs.ucla.edu/~gulzar/rstudio/basic-tutorial.html*](http://web.cs.ucla.edu/~gulzar/rstudio/basic-tutorial.html) *to get a basic understanding of how to load and manipulate data in R.*

In this problem we will create a linear model for predicting the overall happiness score of a nation’s citizens based on a number of societal factors. More specifically, the model will relate:

where are a nation’s scores on a number of societal factors from public health to corruption and is the nation’s predicted happiness score.

Load the world happiness survey data-set into RStudio.

*#path is where the file is located , e.g. “/Users/tylermunger/Documents/tim245/hw2/”*

*data <- read.csv("$PATH$/world\_happiness\_survey\_data.csv")*

*#remove country name from the data-set (we don’t want to use it in the model)*

*data <- data[, !(colnames(data) %in% c("country"))]*

Before creating a linear regression model, it is important to understand of the attributes and instances that are in the dataset. Some basic tools for performing exploratory data analysis in R are shown below:

*#setup attributes*

*attribute\_1 <- data$economy*

*attribute\_2 <- data$family*

*#descriptive statistics*

*mean(attribute\_1)*

*median(attribute\_1)*

*sd(attribute\_1)*

*cor(attribute\_1, attribute\_2)*

*#visual exploration of a single attribute*

*hist(attribute\_1)*

*plot(attribute\_1)*

*boxplot(attribute\_1)*

*#visual exploration of multiple attributes*

*plot(attribute\_1, attribute\_2)*

*boxplot(data)*

*Note: an excellent comprehensive reference for EDA in R can be found at http://r4ds.had.co.nz/exploratory-data-analysis.html*

1. **Use a combination of visual and quantitative tools to answer the following questions for the five input attributes (Xs) and the target (Y):**
2. **What is the typical value (central tendency)?**
3. **What is the uncertainty (spread) for a typical value?**
4. **What is a good distributional fit for the data (symmetric, skewed, long-tailed)?**
5. **Does the attribute affect other attributes (correlation)?**
6. **Does the attribute contain outliers (extreme values)?**

It may be useful to format the answers as a table. Include any relevant plots and descriptive statistics in an appendix section.

1. **Based on the results from the Exploratory Data Analysis, how well does the data-set fit the assumptions of linear regression? For example, is the data normally distributed?**

Create the training and test data-set to use for the linear regression learning algorithm:

*#determine split point for 2/3 train 1/3 test*

*n <- nrow(data)*

*split\_point <- round(n \* 0.66)*

*#randomizes the data-set in order to ensure we get a random sample*

*data <- data[sample(n),]*

*train <- data[1:split\_point,]*

*test <- data[split\_point:n,]*

We will start with the built-in function, lm(), for creating the linear model. The general format for the lm() function is: *lm(target ~ input\_attributes, data=dataset)*.

*#linear model with two attributes: health and family*

*fit.lm <- lm(happiness\_score ~ health + family, data = train)*

*#we can also create a linear model using all of the attributes in the data-set by using a ‘.’*

*fit.lm <- lm(happiness\_score ~ ., data = train)*

*#get the model coefficients*

*coef(fit.lm)*

1. **What is the interpretation of the coefficients of the linear model? For example, which attributes have the strongest correlation with happiness? Does the model make sense?**

Evaluate the performance of the linear model using the test data-set:

*#make predictions*

*predictions.lm <- predict(fit.lm, newdata=test)*

*#summarize accuracy*

*mse.lm <- mean((test$happiness\_score - predictions.lm)^2)*

*mae.lm <- mean(abs(test$happiness\_score - predictions.lm))*

*rae.lm <- mean(abs(test$happiness\_score - predictions.lm) / test$happiness\_score)*

*bias.lm <- mean(predictions.lm - test$happiness\_score)*

*variance.lm <- mean(predictions.lm^2) - mean(predictions.lm)^2*

1. **Provide an assessment of the model’s performance. Do you think the model’s performance is good? Explain why.**

*Note: it may be useful to read the Quick Start section of the documentation of the glmnet package (*[*https://web.stanford.edu/~hastie/glmnet/glmnet\_alpha.html)*](https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html)) *before starting this section of the problem.*

Next, we will use the glmnet package to create regularized linear regression models (ridge, lasso, and elastic net) for predicting happiness. The glmnet package requires that the input data (Xs) and target (Ys) be in matrix form. We therefore need to create separate train and test data-sets as follows:

*#number of attributes including the target y*

*m <- ncol(data)*

*x.train <- as.matrix(data[1:split\_point,1:m-1])*

*x.test <- as.matrix(data[split\_point:n,1:m-1])*

*y.train <- as.matrix(data[1:split\_point,m])*

*y.test <- as.matrix(data[split\_point:n, m])*

The glmnet uses the function glmnet() for creating the linear model. The general format for the glmnet() function is: *lm(x.train, y.train, alpha, lambda).* The parameter alpha is used to control the penalty used in the regularization. Adjusting alpha from 0 to 0.5 to 1 adjusts the regularization from lasso to ridge to elastic net, respectively.

*install.packages("glmnet")*

*library(glmnet)*

*lambda\_parameter <- 0.05*

*fit.lasso <- glmnet(x.train, y.train, alpha=1, lambda=lambda\_parameter)*

*fit.ridge <- glmnet(x.train, y.train, alpha=0, lambda=lambda\_parameter)*

*fit.elnet <- glmnet(x.train, y.train, alpha=.5, lambda=lambda\_parameter)*

*coef(fit.lasso)*

*coef(fit.ridge)*

*coef(fit.elnet)*

1. **How are the coefficients different for the linear model created using lm()?Experiment with lambda values: 0.0005, 0.005, 0.5, and 5. How do the coefficients change as we adjust the value of lambda up and down?**

Evaluate the performance of the linear model using the test data-set:

*#make predictions*

*predictions.lasso <- predict(fit.lasso, x.test, type="link")*

*predictions.ridge <- predict(fit.ridge, x.test, type="link")*

*predictions.elnet <- predict(fit.elnet, x.test, type="link")*

*#summarize accuracy*

*mse.lasso <- mean((y.test - predictions.lasso)^2)*

*mse.ridge <- mean((y.test - predictions.ridge)^2)*

*mse.elnet <- mean((y.test - predictions.elnet)^2)*

*mae.lasso <- mean(abs(y.test - predictions.lasso))*

*mae.ridge <- mean(abs(y.test - predictions.ridge))*

*mae.elnet <- mean(abs(y.test - predictions.elnet))*

*rae.lasso <- mean(abs(y.test - predictions.lasso)/ y.test)*

*rae.ridge <- mean(abs(y.test - predictions.ridge)/ y.test)*

*rae.elnet <- mean(abs(y.test - predictions.elnet)/ y.test)*

*bias.lasso <- mean(predictions.lasso - y.test)*

*bias.ridge <- mean(predictions.ridge - y.test)*

*bias.elnet <- mean( predictions.elnet - y.test)*

*variance.lasso <- mean(predictions.lasso^2) - mean(predictions.lasso)^2*

*variance.ridge <- mean(predictions.ridge^2) - mean(predictions.ridge)^2*

*variance.elnet <- mean(predictions.elnet^2) - mean(predictions.elnet)^2*

Please answer the following question:

1. **Which model would you recommend to a nation that is trying to predict the happiness of their citizens?**
2. **What is the interpretation of the selected model? What can we say about the relationship between the input attributes and the happiness score?**

**Extra Credit:**

**Show that coef(lm(Y ~ X)) is equivalent to (hint: remember that you will need to add a leading column of “1”s to the matrix X).**

**Problem 2: Model Selection in Weka for Predicting Customer Churn**

In this problem you will create a classification model that can determine if a particular customer at a large telecommunications company will stop paying for service, i.e. churn. More specifically, the model will take in basic information about any particular customer and produce a prediction for if the customer will stop service:

where are demographic information and services that the customer is consuming and is the binary prediction {yes, no} for churn.

There are a variety of different learning algorithms that can be used to create the churn classification model. These algorithms have certain general characteristics and we can make hypotheses of which algorithm will perform well on for particular dataset. However, in complex real-world data-sets it is difficult to reliably determine which algorithm will produce the best model without going through the process of empirical evaluation.

The purpose of this problem is to get experience going through the so-called model selection process by building and testing a variety of different models in order to find the one that is the best fit for the data-set under consideration. To this end, you will be performing four experiments using the following learning algorithms:

1. Bayesian: Naïve Bayes
2. Functional: Logistic Regression, Support Vector Machines (SMO)
3. Lazy: K-Nearest Neighbor (IBK)
4. Rules: ZeroR
5. Trees: J48, Random Forest

Use 10-fold cross validation and the default parameters for all the classifiers in Weka, unless specified otherwise. Your evaluation of the models in each experiment should be based on a combination of both accuracy and f-measure.

Experiment 1: Establish a baseline for the classification model performance using the following learning algorithms: ZeroR, Naïve Bayes, and Decision Tree (J48).

1. **How does the ZeroR model work? How is the ZeroR useful when creating a baseline?**
2. **Compare and contrast the performance of the three models. Do the Naïve Bayes and Decision Tree models significantly outperform the ZeroR model? Explain why or why not.**

Experiment 2: Determine the performance of a simple non-linear model using the K-Nearest Neighbors algorithm.

1. **How does the performance of the K-NN model compare to the baseline models from Experiment 1?**
2. **Should the input attributes be normalized? Explain why or why not.**
3. **Experiment with the following values for K=3,10,50,100. What effect does changing K have on the model performance? What is the optimal value for K? Explain why.**

Experiment 3: Determine the performance of a linear model using the Logistic Regression and Support Vector Machine algorithms

1. **Compare and contrast the linear models (Logistic Regression and SVM) to the baseline and the non-linear models from the previous experiments?**
2. **Which model performs better: logistic regression or SVM? If there is a difference, explain why.**

Experiment 4: Determine the performance of an ensemble model using the Random Forest algorithm.

1. **Compare and contrast the Random Forest model with the results from Experiments 1, 2, and 3.**
2. **What is the difference between a Random Forest and the decision tree model from Experiment 1. Would you expect the Random Forest model to outperform the decision tree model? Explain why.**

Based on the results from Experiments 1-4, please answer the following questions:

1. **What model do you recommend that the telecommunications company use to predict churn? Explain why.**
2. **What are the limitations of the recommended model? How do you recommend that it be used?**

**Extra credit**

**Perform an additional experiment using a learning algorithm of your choice. Compare and contrast the performance with the previous experiments and explain the reason(s) for any differences.**