**IE 425**

**Homework 2 (due May 14 23:59pm)**

1. Consider the “credit.csv” file where **balance** (average credit card debt for a number of individuals) is the output attribute. Quantitative input attributes are age, cards (number of credit cards), education (years of education), income (in thousands of dollars), limit (credit limit), and rating (credit rating). Qualitative input attributes are gender, student (student status), status (marital status), and ethnicity (Caucasian, African American or Asian).

(a) Using linear regression, investigate whether gender has an effect on the credit card balance. What is the average credit card debt for males and females?

(b) Using linear regression, investigate whether ethnicity has an effect on the credit card balance.

(c) We want to investigate whether the effect of income on balance is different for a student compared to a non-student. How can we do this?

(d) Perform linear regression first with input attributes “age” and “limit”, and then with “rating” and “limit”. What are your conclusions?

2. Using the “UniversalBank.csv” file, split the data set into training (80%) and test (20%) sets with a seed value of “4250”. Apply *k*-fold cross-validation method with *k*=10 on the training set to find the best odd value of *k* (between 1 and 11) to be used in the *k*-NN algorithm. Then report the confusion matrix on the test set using the best *k* value (Use the caret package)

3. When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is

known as the curse of dimensionality, and it ties into the fact that curse of dinon- parametric approaches often perform poorly when p is large. We mensionality will now investigate this curse.

(a) Suppose that we have a set of observations, each with measurements on p = 1 feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation’s response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with X = 0.6, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?

(b) Now suppose that we have a set of observations, each with measurements on p = 2 features, X1 and X2. We assume that (X1,X2) are uniformly distributed on [0, 1] × [0, 1]. We wish to predict a test observation’s response using only observations that are within 10% of the range of X1 and within 10% of the range of X2 closest to that test observation. For instance, in order to predict the response for a test observation with X1 = 0.6 and

X2 = 0.35, we will use observations in the range [0.55, 0.65] for X1 and in the range [0.3, 0.4] for X2. On average, what fraction of the available observations will we use to make the prediction?

(c) Now suppose that we have a set of observations on p = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation’s response using observations within the 10% of each feature’s range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

(d) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations “near” any given test observation.

(e) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For p = 1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer.

Note: A hypercube is a generalization of a cube to an arbitrary number of dimensions. When p = 1, a hypercube is simply a line segment, when p = 2 it is a square, and when p = 100 it is a 100-dimensional cube