The Analytics Edge

Test Your Knowledge of Clustering and Recommendation Systems

1. In this question, we will use cluster-then-predict, a methodology in which you first cluster observations and then build cluster-specific prediction models. In this assignment, we'll use cluster-then-predict to predict future stock prices using historical stock data. When selecting which stocks to invest in, investors seek to obtain good future returns. In this problem, we will first use clustering to identify clusters of stocks that have similar returns over time. Then, we'll use logistic regression to predict whether or not the stocks will have positive future returns. For this problem, we'll use StocksCluster.csv, which contains monthly stock returns from the NASDAQ stock exchange. The NASDAQ is the second-largest stock exchange in the world, and it lists many technology companies. The stock price data used in this problem was obtained from infochimps, a website providing access to many datasets. Each observation in the dataset is the monthly returns of a particular company in a particular year. The years included are 2000-2009. The companies are limited to tickers that were listed on the exchange for the entire period 2000-2009, and whose stock price never fell below $1. So, for example, one observation is for Yahoo in 2000, and another observation is for Yahoo in 2001. Our goal will be to predict whether or not the stock return in December will be positive, using the stock returns for the first 11 months of the year. This dataset contains the following variables:

   - **ReturnJan:** the return for the company's stock during January (in the year of the observation)
   - **ReturnFeb:** the return for the company's stock during February (in the year of the observation)
   - **ReturnMar:** the return for the company's stock during March (in the year of the observation)
   - **ReturnApr:** the return for the company's stock during April (in the year of the observation)
   - **ReturnMay:** the return for the company's stock during May (in the year of the observation)
   - **ReturnJune:** the return for the company's stock during June (in the year of the observation)
   - **ReturnJuly:** the return for the company's stock during July (in the year of the observation)
   - **ReturnAug:** the return for the company's stock during August (in the year of the observation)

- **ReturnSep:** the return for the company's stock during September (in the year of the observation)
- **ReturnOct:** the return for the company's stock during October (in the year of the observation)
- **ReturnNov:** the return for the company's stock during November (in the year of the observation)
- **PositiveDec:** whether or not the company's stock had a positive return in December (in the year of the observation). This variable takes value 1 if the return was positive, and value 0 if the return was not positive.

For the first 11 variables, the value stored is a proportional change in stock value during that month. For instance, a value of 0.05 means the stock increased in value 5% during the month, while a value of -0.02 means the stock decreased in value 2% during the month.

(a) Load **StocksCluster.csv** into a data frame called **stocks**. How many observations are in the dataset?

(b) What proportion of the observations have positive returns in December?

(c) What is the maximum correlation between any two return variables in the dataset? You should look at the pairwise correlations between ReturnJan, ReturnFeb, ReturnMar, ReturnApr, ReturnMay, ReturnJune, ReturnJuly, ReturnAug, ReturnSep, ReturnOct, and ReturnNov

(d) Which month (from January through November) has the largest mean return across all observations in the dataset? Which month (from January through November) has the smallest mean return across all observations in the dataset?

(e) Run the following commands to split the data into a training set and testing set, putting 70% of the data in the training set and 30% of the data in the testing set:

```
> set.seed(144)
> spl <− sample.split(stocks$PositiveDec, SplitRatio = 0.7)
> stocksTrain <− subset(stocks, spl == TRUE)
> stocksTest <− subset(stocks, spl == FALSE)
```

Then, use the stocksTrain data frame to train a logistic regression model (name it StocksModel) to predict PositiveDec using all the other variables as independent variables. What is the overall accuracy on the training set, using a threshold of 0.5?

(f) Now obtain test set predictions from StocksModel. What is the overall accuracy of the model on the test, again using a threshold of 0.5?

(g) What is the accuracy on the test set of a baseline model that always predicts the most common outcome in the training set?

(h) Now, let's cluster the stocks. The first step in this process is to remove the dependent variable using the following commands:

```
> limitedTrain <− stocksTrain
```

> limitedTrain$PositiveDec <− NULL
> limitedTest <− stocksTest
> limitedTest$PositiveDec <− NULL

Why do we need to remove the dependent variable in the clustering phase of the cluster-then-predict methodology?

i. Leaving in the dependent variable might lead to unbalanced clusters
ii. Removing the dependent variable decreases the computational effort needed to cluster
iii. Needing to know the dependent variable value to assign an observation to a cluster defeats the purpose of the methodology

(i) In some cases where we have a training and testing set, we might want to normalize by the mean and standard deviation of the variables in the training set. We can do this by using the **caret** package passing just the training set to the preProcess function, which normalizes variables by subtracting by the mean and dividing by the standard deviation.
> library(caret)
> preproc <− preProcess(limitedTrain)
> normTrain <− predict(preproc, limitedTrain)
> normTest <− predict(preproc, limitedTest)
What is the mean of the ReturnJan variable in normTrain?
What is the mean of the ReturnJan variable in normTest?

(j) Why is the mean ReturnJan variable much closer to 0 in normTrain than in normTest?

i. Small rounding errors exist in the normalization procedure
ii. The distribution of the ReturnJan variable is different in the training and testing set
iii. The distribution of the dependent variable is different in the training and testing set

(k) Set the random seed to 144 (it is important to do this again, even though we did it earlier). Run k-means clustering with 3 clusters on normTrain, storing the result in an object called km. Which cluster has the largest number of observations?

i. Cluster 1
ii. Cluster 2
iii. Cluster 3

(l) In this question, we use the flexclust package to obtain training set and testing set cluster assignments for our observations and to do the predictions. Use the following commands:
> library(flexclust)
> km.kcca <− as.kcca(km, normTrain)
> clusterTrain <− predict(km.kcca)
> clusterTest <− predict(km.kcca, newdata=normTest)
How many test-set observations were assigned to Cluster 2?

(m) Using the subset function, build data frames stocksTrain1, stocksTrain2, and stocksTrain3, containing the elements in the stocksTrain data frame assigned to clusters 1, 2, and 3, respectively (be careful to take subsets of stocksTrain, not of normTrain). Similarly build stocksTest1, stocksTest2, and stocksTest3 from the stocksTest data frame. Which training set data frame has the highest average value of the dependent variable?

   i. stocksTrain1
   ii. stocksTrain2
   iii. stocksTrain3

(n) Build logistic regression models StocksModel1, StocksModel2, and StocksModel3, which predict PositiveDec using all the other variables as independent variables. StocksModel1 should be trained on stocksTrain1, StocksModel2 should be trained on stocksTrain2, and StocksModel3 should be trained on stocksTrain3. Which variables have a positive sign for the coefficient in at least one of StocksModel1, StocksModel2, and StocksModel3 and a negative sign for the coefficient in at least one of StocksModel1, StocksModel2, and StocksModel3?

(o) Using StocksModel1, make test-set predictions called PredictTest1 on the data frame stocksTest1. Using StocksModel2, make test-set predictions called PredictTest2 on the data frame stocksTest2. Using StocksModel3, make test-set predictions called PredictTest3 on the data frame stocksTest3.
What is the overall accuracy of StocksModel1 on the test set stocksTest1, using a threshold of 0.5?
What is the overall accuracy of StocksModel2 on the test set stocksTest2, using a threshold of 0.5?
What is the overall accuracy of StocksModel3 on the test set stocksTest3, using a threshold of 0.5?

(p) To compute the overall test-set accuracy of the cluster-then-predict approach, we can combine all the test-set predictions into a single vector and all the true outcomes into a single vector:
```
> AllPredictions <− c(PredictTest1, PredictTest2, PredictTest3)
> AllOutcomes <− c(stocksTest1$PositiveDec, stocksTest2$PositiveDec, stocksTest3$PositiveDec)
```
What is the overall test-set accuracy of the cluster-then-predict approach, again using a threshold of 0.5?

2. Clustering is commonly used to divide a broad target market of customers into smaller, similar groups and then to design marketing strategies specifically for each group. In this question, you will use clustering to study publicly available data from the New York Citi Bike sharing program. Citi Bike is the largest bike sharing program in the United States and serves various parts of the New York city. The data is provided in the file **citibike.csv** and contains trip information for the bike rides for the month of July 2013:

   - **tripduration**: Time duration of the trip (in seconds)
   - **startstation**: Name of the start station

- **endstation**: Name of the end station
- **gender**: Gender of user (1 = male, 2 = female)
- **age**: Age of the user
- **day**: Day on which the trip was started (Mon, Tue, Wed, Thu, Fri, Sat, Sun)
- **starttime**: Start time of the trip in unit of hours (measured from 0 (12am) to 23 (11 pm))

(a) Read the dataset into the dataframe **citi**. How many bike stations are there in this dataset?

(b) On which day of the week, is the average duration of the trips taken by bikers the maximum?

(c) What is the start hour when the maximum number of bikes are rented? What is the start hour when the minimum number of bikes are rented?

(d) In this dataset, what proportion of the bikes are rented by female users?

(e) One of the challenges to do clustering with this data is the presence of the categorical variable for the **day** variable. To tackle this, we will define seven new binary variables (**Mon** to **Sun**), each of which takes a value of 1 if the corresponding trip is taken on that day and 0 otherwise. Write down one such sample R command(s) that you use to do this for a given day of the week.

(f) In clustering data, it is often important to normalize the variables so that they are all on the same scale. If you clustered this dataset without normalizing, which variable would you expect to dominate in the distance calculations?

- **tripduration**
- **gender**
- **age**
- **starttime**
- **Mon**

(g) Normalize the variables **tripduration**, **gender**, **age**, **starttime**, **Mon**, ..., **Sun** by using the **scale()** function in R. We normalize such that each variable has mean 0 and standard deviation 1. What is the maximum value of **tripduration** in the normalized dataset?

(h) We will not use hierarchical clustering for this dataset. Why do you think hierarchical clustering might have a problem with this dataset?

- We have categorical variables in this dataset, so we cant use hierarchical clustering.
- We might have too many variables in the dataset for hierarchical clustering to handle.
- We might have too many observations in the dataset for hierarchical clustering to handle.
- We are sure of the number of clusters in this application, so using hierarchical clustering does not make sense.

(i) Run the k-means algorithm on the normalized dataset with 10 clusters. Use set.seed(100) in R just before running the code. You only want to use the variables **tripduration**, **gender**, **age**, **starttime**, **Mon**, ..., **Sun** in building your clusters. Use the default settings to build your model. How many trips are there in the largest and smallest clusters respectively?

(j) Which cluster best fits the description "trips taken primarily by older users on Saturdays"? You can use the centers of the clusters to answer this question.

(k) Which cluster best fits the description "longer trips taken primarily by female users either on Tuesdays or Wednesdays"? You can use the centers of the clusters to answer this question.

(l) If we ran k-means clustering a second time without making any additional calls to set.seed, we would expect

- Different results from the first k-means clustering
- Identical results to the first k-means clustering

(m) If we ran k-means clustering a second time, again running the command set.seed(100) right before doing the clustering, we would expect

- Different results from the first k-means clustering
- Identical results to the first k-means clustering

(n) Suppose the marketing department at Citi Bike decided that instead of using the days of the week for clustering, they would like to use a single variable **weekday** which took a value of 1 if the trip started on Monday, Tuesday, Wednesday, Thursday, or Friday and 0 if it started on a Saturday or Sunday. Redo the clustering. As before, remember to normalize the **weekday** variable and run the k-means algorithm on the normalized dataset with 10 clusters. Use set.seed(100) in R before running the code. Which cluster best fits the description "longer trips taken by older female users on weekdays"? You can use the centers of the clusters to answer this question.

(o) Which cluster best fits the description "short trips taken by younger male users early on weekdays"? You can use the centers of the clusters to answer this question.

3. In this question, you will extend the collaborative filtering model from the class. In the class we developed three models for collaborative filtering—using average item rating, using average user rating and by taking the average of the k nearest users using the Pearson correlation metric. In this question, you will extend the last model by using a weighted average where the weights are the similarity metric defined by the Pearson correlation. Develop an R code to do this and verify the quality of the fit by changing the number of neighbors in the set $\{10, 50, 100, 150, 200, 250\}$. Compute the root mean squared error in each of these cases.