Lab 7: Machine Learning – Clustering and MariaDB (MySQL)

|  |  |
| --- | --- |
| Due: | Wednesday, July 1st at the end of class |
| Assignment Type: | Group (Up to 5 – may be randomly assigned) |
| Assignment Title: | Introduction to Machine Learning – Clustering |
| Style: | One problem at a time on Discord – WORK TOGETHER |
| External Sources: | You are allowed notes, books, and searches |
| Description: | We are going to do assignments dealing with Clustering and relational databases. |
| Points | 50 |
| Starting Files | On Blackboard (Download there) |

Enter your group member names here:

You will complete the first two problems using R and then the last four questions using XAMPP.

This lab was introduced by Jeremy Dixon and based on <https://www.edureka.co/blog/k-means-clustering-algorithm/>

# Using Clustering in R

1. You need to download the file **movies.csv** for this problem. Download it from Blackboard.

For this problem, we are going to be working through an example of k-means to be able to predict gross related to budget.

You will need to create a new project and add the movies.csv file to that new folder. In addition, you will need to update the code below to use your path instead of mine. You can check your progress at the bottom of this page: <https://www.edureka.co/blog/k-means-clustering-algorithm/>

The data in **movies.csv** can be summarized by using: summary(movies) once it has been loaded.

For this, we will need the following packages. You should only need to load them once so after you install all of them, you can remove this part.

install.packages('tidyverse')

install.packages('ggplot2')

library(tidyverse) #data science library

library(ggplot2) #plotting library

setwd("~/UMBC/2024/Summer/CMSC491/R/clustering/lab6") #working directory

movies <- read\_csv("movies.csv") #load csv file

movie <- data.matrix(movies) #convert dataframe to matrix

movie <- na.omit(movie) #remove NA from dataset

smple <- movie[sample(nrow(movie),500),] #Only take the top 500 rows

smple\_short <- subset(smple, select=c(9,23)) #Only take the gross and budget columns

smple\_matrix <- data.matrix(smple\_short) #Convert to matrix

View(smple\_matrix) #View the Matrix

wss <- (nrow(smple\_matrix)-1)\*sum(apply(smple\_matrix,2,var))#Determines number of clusters and shows elbow plot

for (i in 2:15) wss[i]<-sum(kmeans(smple\_matrix,centers=i)$withinss)

plot(1:15, wss, type="b", xlab="Number of Clusters", ylab="Within Sum of Squares")

#As you can see, there is a sudden drop in the value of WSS (within sum of squares)

#as the number of clusters increase from 1 to 3.

#Therefore, the bend at k=3 gives the stability in the value of WSS.

#We need to strike a balance between k and WSS.

#So, in this case, it comes at k=3.

#Apply inbuilt kmeans function in R to form clusters.

cl <- kmeans(smple\_matrix,3,nstart=25)

plot(smple\_matrix, col =(cl$cluster +1) , main="k-means result with 2 clusters", pch=1, cex=1, las=1)

points(cl$centers, col = "black", pch = 17, cex = 2)

#Analyze how good is the cluster formation

cl

#Here, total\_SS is the sum of squared distances of each data point to the global sample mean whereas

#between\_SS is the sum of squared distances of the cluster centroids to the global mean.

#Here, the percentage is a measure of the total variance in the data set.

#The goal of k-means is to maximize the between-group dispersion(between\_SS).

#So, higher the percentage value, better is the model.

#For a more in-depth look at the clusters, we can examine the coordinates of the

#cluster centroids using the cl$centers component, which is as follows for

#gross and budget (in million).

cl$centers

Show us your lab7\_1.R file and the plots.

# Using Clustering in R

1. For this problem, you are going to use a dataset that you find to repeat the process above. You can choose to use other facets of the movies database or you can find a new one. I will be impressed if you can find a new dataset to use that shows us something interesting!

Show us the dataset file and your lab7\_2.R file including the plots.

# XAMPP

1. The first part of this lab is to load XAMPP onto your computer. It is a free piece of software that is an open-source solution for hosting websites on your local machine. It provides Apache, MariaDB (which is a version of mySQL), PHP, and Perl. We will use it for the relational database parts of this class.

XAMPP installation Instructions:

Go to:

<https://www.apachefriends.org/index.html>

Click on the version for your local machine. I am using a Windows based machine so I will show you that. There is a version for Linux, Mac, and Windows.

Go to the location where you installed XAMPP (usually **C:\Program Files\xampp**) and double click on XAMPP Control Panel (**xampp-control.exe**). This will bring you following screen. Click on **Start** buttons next to Apache and MySQL for starting them.

A screenshot of a computer

Description automatically generated

## Apache Is Not Starting Error

Sometimes you would experience that even after clicking Start button several times, Apache is not starting. This is usually because some other service is running at the port required by Apache which is 80 by default. An easy reproducible way for this error is starting [Skype](http://www.skype.com/) before starting Apache.

In such a case, you would need to stop other service temporary and restart it after starting Apache. For an example, if you do so for Skype, it will find another port for working after restarting.

## Making Requests to the Server

Once you started Apache in control panel, type **http://localhost** in your web browser. This would bring you a web page that lists XAMPP related details.

## Putting Stuff in Web Folder

Under XAMPP root directory there is a folder called **htdocs**. That’s where you should put your web site related stuff. For each web site you create, it’s better to create a folder inside **htdocs** folder and then put content inside that to avoid conflicts.

You can access the dashboard via the URL <http://localhost/dashboard/>

Everyone in the group should install XAMPP.

# Data Cleanup in Excel

1. You need to download the file farmers\_markets.xlsx for this problem. Download it from Blackboard.

Initially, we need to clean up the data. There are some duplicates. Some without zip codes and some that are missing a significant amount of data for what they sell (which are Boolean).

For this problem complete the following:

* Add zip codes for all missing locations based on their city and state.
* If there is a product (such as Coffee, Beans, Fruits, or Grains) listed that is “blank” assume that it does not sell it and make it a “N”
* Remove any with no city or state
* Make sure that all state data is consistent (either abbreviation or full name)
* Try to remove duplicates (Hard to do – but possible) (Not graded)

Show us the cleaned data in lab7\_4.xlsx.

# Splitting into Tables

1. Create a new database named “**CMSC462\_lab7**”. For this problem, import the farmers\_market table into MySQL as a table named “markets”. You will probably first need to save it as a CSV. They you are going to write queries to create new tables splitting the farmers\_market tables into three smaller tables. Keep track of each of the queries that you have used to manipulate the data below.
   * Give a primary key to the “markets” table.
   * Make a table named “location” for all unique cities, states, and zip codes. Give this table a locationID that is the primary key. You can choose how to do this by using either city/state or zip or some combination. Add the locationID to each market and remove the city, state and zip from the “markets” table.
   * Make a new table named “websites” for each market that has a website. Store just the market’s primary key and the website. Remove the website field from “markets”
   * Make a new table named “products” that lists what products each market sells (as Booleans). Store the market’s primary key and each product as a Boolean. Remove all of the products from “markets”

In the end, you should have the markets table (with the locationID and missing the websites and all of the product fields). You should have a websites table (with just the marketID and website). You should have a products table (with the marketID and the products).

Show us the updated database in phpMyAdmin and the list of queries used to manipulate the data.

# Writing a Multi-Table Query

1. For this problem, write a query that lists all farmers markets from California.

Save the query as lab7\_6.sql and show it to us.