DAEN 500-1 – Data Analytics Fundamentals

Spring 2021 Final Examination Exercise

3/5 – 3/6/2021

Final Submission Deadline: NLT 11:59PM (EST). Saturday, March 6, 2010

*Failure to submit ON TIME will result in DAEN COURSE FAILURE*

Name:\_\_\_\_\_Timothy Dugan\_\_\_\_\_\_\_\_\_\_\_\_\_\_ GMU G#\_\_0 00492781\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Signature (Honor Certification): \_\_\_\_Timothy Dugan\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This exam is **OPEN BOOK/OPEN NOTES**. You may consult any of the course texts, and the various reference materials recommended in the syllabus. ***The exam of course IS NOT “Open Web”,*** especially in that you may NOT utilize expert “help” sites such as Stack Overflow, or other programming help or collaboration sites.

HONOR CODE CERTIFICATION

**Your signature above declares that you have followed the conditions of this exam, and that the work is yours alone**. **Specifically:**

This must be your own work, authored and completed by you. As stated earlier, this is an “open source exam” – allowing books, notes or courseware, as well as *general* expert advice gained PRIOR to exam. YOU MAY NOT, HOWEVER, SEED OR USE ANY ADVICE ON HOW TO SOLVE THE QUESTION OR ANY CODE WRITTEN BY ANY OTHER INDIVIDUAL. *Any violation will result in an immediate failure in the exam and for the course, as well as referral to the GMU Honor Committee for determination of any other appropriate disciplinary consequences.*

*NOTE: Your* ***submission*** *of any responses, files, programs, etc. in response to the DAEN500 final exam instructions, will also be your personal certification of your full compliance with the spirit and letter of the* ***GMU Honor Code*** *standards for take home and/or in-class exams.*

Additionally, you are restricted from discussing the substance of the questions on this exam with any other individual, until after you have submitted your final response for grading. The completed exam -- with your answers embedded in this docx document (add extra pages as necessary) should be submitted following instructions contained in the Final Exam Instructions BB site. If you have any trouble submitting and have extra parts of the answers you have trouble appending to this document, you may simply submit additional pages separately (the exam submission site is set for multiple submissions, just in case). Make certain all are submitted PRIOR TO THE DEADLINE!

 FINAL EXAM PROBLEMS

COMPLETE ALL & INSERT ANSWERS BELOW QUESTIONS

# Problem 1: Python Programming Problem (15 Points Total)

* **Design and implement a Python program that is based on the following requirements: a) program will find all numbers which are divisible by 7 but are not a multiple of 5; and b) numbers between 2000 and 3200.**
* **INSERT (cut&paste) your Python code in space below and *then insert a screen shot in space below, showing code, your successful run, input and output.***

NOTE of alternative for help: To help test your code, you also may use a Python “programming window” found in the **Zybooks Section 35 Additional Material** OR any relevant IDE.

#find all numbers which are divisible by 7 but are not multiples of 5  
# and are numbers between 2000 and 3200  
  
number\_space = range(2000,3200)  
div7less5 = []  
for n in number\_space:  
 #nested if statements in loop iterating for every number between 2,000 and 3,200  
 if n % 5 != 0:  
 if n % 7 == 0:  
 #adds number to array if qualifying  
 div7less5.append(n)  
  
#outputs array for validation  
[2002, 2009, 2016, 2023, 2037, 2044, 2051, 2058, 2072, 2079, 2086, 2093, 2107, 2114, 2121, 2128, 2142, 2149, 2156, 2163, 2177, 2184, 2191, 2198, 2212, 2219, 2226, 2233, 2247, 2254, 2261, 2268, 2282, 2289, 2296, 2303, 2317, 2324, 2331, 2338, 2352, 2359, 2366, 2373, 2387, 2394, 2401, 2408, 2422, 2429, 2436, 2443, 2457, 2464, 2471, 2478, 2492, 2499, 2506, 2513, 2527, 2534, 2541, 2548, 2562, 2569, 2576, 2583, 2597, 2604, 2611, 2618, 2632, 2639, 2646, 2653, 2667, 2674, 2681, 2688, 2702, 2709, 2716, 2723, 2737, 2744, 2751, 2758, 2772, 2779, 2786, 2793, 2807, 2814, 2821, 2828, 2842, 2849, 2856, 2863, 2877, 2884, 2891, 2898, 2912, 2919, 2926, 2933, 2947, 2954, 2961, 2968, 2982, 2989, 2996, 3003, 3017, 3024, 3031, 3038, 3052, 3059, 3066, 3073, 3087, 3094, 3101, 3108, 3122, 3129, 3136, 3143, 3157, 3164, 3171, 3178, 3192, 3199]print(div7less5)

# Problem 2: Python Programming Problem

# (15 Points Total)

* **Design and implement a Python program that is based on the following requirements:**

**a) define a class which has *at least two* methods**

* + **Method 1 – getString: to get a string from console input; and,**
  + **Method 2 - printString: to print the string in upper case.**

**b) demonstrate code works using three different test input strings**

* ***INSERT* *code below* and *INSERT* a screen shot of the program and successfully run output that *includes test input for input strings (test strings must include (a) all upper case, (b) all lower case, and (c) mix of upper and lower case).***

#Design and implement a Python program that is based on these requirements:  
# a) define a class which has at least two methods  
# o Method 1 – getString: to get a string from console input; and,  
# o Method 2 - printString: to print the string in upper case.  
# b) demonstrate code works using three different test input strings  
  
#defintes relay class with default constuctor taking no args  
class relay:  
 def \_\_init\_\_(self):  
 #instance var: intake is an empty string  
 self.intake = ''  
  
 def getString(self):  
 #reassigns instance var to return of input()  
 self.intake = input()  
 def printString(self):  
 #relays instance variable via print method  
 print(self.intake)  
  
#demonstration  
# test\_conditions:'abcdefghijk' 'ABCDEFGHIJK' 'aAbBcCdDeEfFgGhHiIjJkK'  
  
listner = relay()  
listner.getString()  
listner.printString()

"Z:\Users\Tim User Level\PycharmProjects\pythonProject\venv\Scripts\python.exe" "Z:/Users/Tim User Level/PycharmProjects/pythonProject/Problem2.py"

abcabc ABCABC AbCAbC

abcabc ABCABC AbCAbC



# Problem 3: R Programming Problem

# (20 Points Total)

* **Perform the following problems using R:**
  + Create a vector of courses (e.g., MATH 101) you have taken previously. Make sure you have at least 8 courses. Name the vector myCourses
  + Get the length of the vector myCourses
  + Get the first two courses from myCourses
  + Get the 3rd and 4th courses from myCourses
  + Sort myCourses using a method
  + Sort myCourse in the reverse direction
* *INSERT* *code below* and *INSERT* a screen shot of the program and successfully run output.

**myCourses <- c("ECON 101", "MATH 102", "ART200", "ENGLISH 203", "MATH 230", "CHEM 150", "DAEN 500", "STAT 200")**

**transcript\_size <- length(myCourses)**

**first\_pair <- myCourses[c(1,2)]**

**second\_pair <- myCourses[c(3,4)]**

**myCourses\_alpha <- sort(myCourses)**

**myCourses\_backwards <- sort(myCourses,decreasing =TRUE)**



# Problem 4: Principal Component Analysis

# (25 points)

**Provide a description of the following:**

1. What is a component – Provide a description (5 points)
2. Principal Component Analysis – Provide a description.(5 points)
3. **Provide an specific example of Principal Component Analysis(15 points)**

**Answer:**

A component is a variable created from multiple independent variables typically written in the form “Zsubi” where ‘i’ is the components order(based on its eigenvalue) for the permutation of a predictive model which utilizes PCA

Components in relation to PCA are used to reduce the dimensional complexity of a system of data points while preserving as much information as possible. It does this by combining correlated data and collapsing it into components with respective eigenvalue/vector pairs that corresponds to a components variance in terms of direction and magnitude. When eigenvalues(variance) are ordered from greatest to least the principal components are arranged by significance in representing the sample data. This elucidates the importance of a given data subsets(component) . Knowing this significance valuable information loss can be minimized while reducing datasets into more discernable formats.

We will first find the eigenvalue of a 2x2 matrix with values representing books read and hours tv watched transformed into a matrix.

1. First we set the matrix equal to – lambda( eigenvalue) \* the identiy matrix = 0

2 Solve for lambda . 3 reduces via the characteristic equation to (1-lambda)^2 = (0.4)^2

3 finally you get the eigenvalues 1. 4 and 0.6

Then plug in the eigenvalues and solve for the eigenvectors identity = eigenvalue\*

Using the values 1.4 and 0.6 we find two equations and then balance them to determine that v1,1 = v1,2

Hence the first eigenvector is and

Create the principal component for 2 dimensions create Zsub1 = Vsub1,1Xsub1 + Vsub1,2Xsub2 = Xsub1 + Xsub2

Zsub2 = Vsub2,1Xsub1 + Vsub2,2Xsub2 = -Xsub1 + Xsub2

In this case Zsub1 has the larger eigenvalue and will be largely more important in terms of its weight on the outcome of a dependent variable

# Problem 5: Multiple vs. Logistic

# (30 points)

# Describe: What is difference between Multiple Regression and Logistic Regression? What circumstances might determine which to use? (10 points)

# Demonstrate: Using any data, and any tool set you’ve learned about, show differences (20 points)

# SUGGESTION: may be solved using RapidMiner, or other toolsets, BOTH TO ANALYZE AND TO VISUALIZE REGRESSION DIFFERENCES.

Step 1: Perform a quick search of the [UCIS public data archive](https://archive.ics.uci.edu/), a well-curated site which you already have seen as part of your introductory RapidMiner training.

Step 2: Pick a dataset you find interesting, input dataset into regression tools you’ve chosen.

Step 3: Run regression, .and use visualizations to demonstrate the conceptual answers you provided for 5.(a).

A) Both of these regression types use labeled data to find trends and make predictions. The main difference is how they are applied for different goals.

Multiple linear regressions are used for continuous data and involve more than one predictor variable to fit a line through a trend that most accurately captures real values of a population given a sample. It also factors in a residual error which is the difference in population dependent variables and the predicted one.

Logistic regressions use a binary response variable for classification questions.

EX. Is obese = 1 not obese =0 Also it uses a different function that creates an S curve between the 0 and 1 point. We predict the results of categorical variables.

