DAEN 500- DL1 – Data Analytics Fundamentals

Fall 2020 Final Examination Exercise

11/24 – 12/05/2020

Final Submission Deadline: NLT 11:59PM (EST). Saturday, Dec 5, 2010

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

Name:\_\_ STAVROS KALAMATIANOS \_\_\_\_\_\_\_\_\_\_\_\_\_ GMU G#\_\_\_G00022302\_\_\_\_\_

Student Signature (Honor Certification): \_\_\_\_\_\_\_ STAVROS KALAMATIANOS\_\_\_\_\_\_\_\_\_

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.***

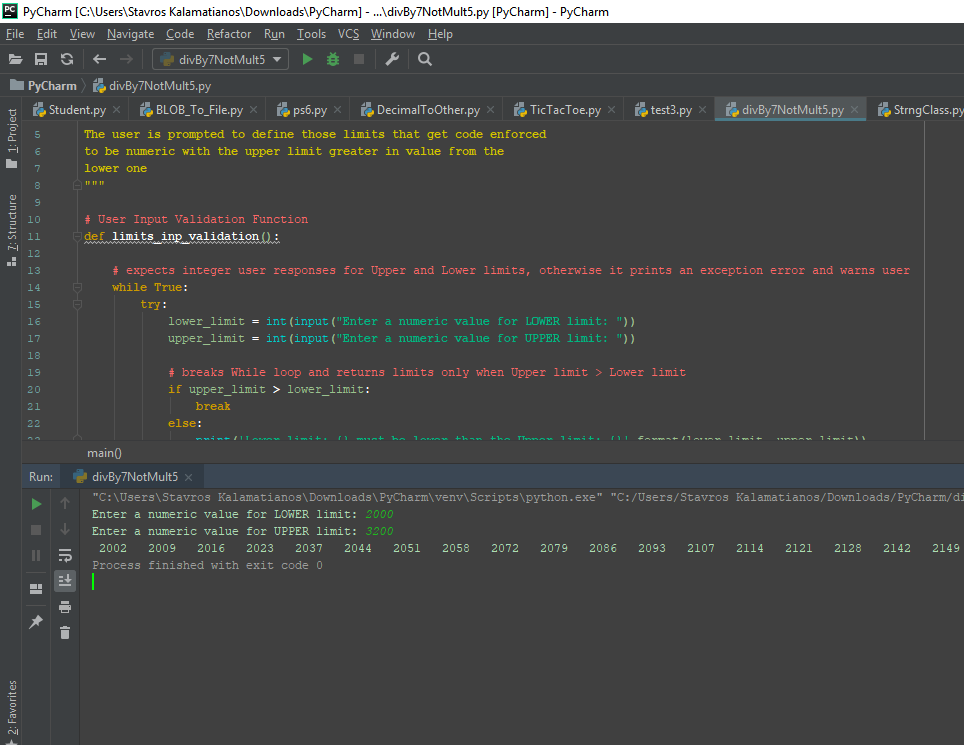
NOTE of alternative for help: To help test your code, you also may use a Python “programming window” found in the. **Zybooks Section 35 Additional Material**.

**Response to Problem 1:**

The Python Script was written in PyCharm Community edition 2018.3 per PEP-8 formatting rules.

**"""  
This Python script returns all numbers divisible by 7 that  
are not multiples of 5 between an upper and lower integer limit  
  
The user is prompted to define those limits that get code enforced  
to be numeric with the upper limit greater in value from the  
lower one  
"""**# User Input Validation Function  
**def limits\_inp\_validation():** # expects integer user responses for Upper and Lower limits, otherwise it prints an exception error and warns user  
 **while True:  
 try:** lower\_limit **=** int**(**input**("Enter a numeric value for LOWER limit: "))** upper\_limit **=** int**(**input**("Enter a numeric value for UPPER limit: "))** # breaks While loop and returns limits only when Upper limit > Lower limit  
 **if** upper\_limit **>** lower\_limit**:  
 break  
 else:** print**('Lower limit: {} must be lower than the Upper limit: {}'**.format**(**lower\_limit, upper\_limit**))  
  
 except** Exception **as** e**:** print**(" Upper and lower limits must be integers, please try again ")** print**(**e**)** print**()  
  
 return** lower\_limit, upper\_limit  
  
  
**def main():** l, u **=** limits\_inp\_validation**()** result **= False  
 for** i **in** range**(**l, u**):  
  
 if (**i **%** 7 **==** 0**) and (**i **%** 5 **!=** 0**):** print**(' {} '**.format**(**i**)**, end**=" ")** result **= True  
  
 if not** result**:** print**("no numbers found")  
  
  
if** \_\_name\_\_ **== "\_\_main\_\_":** main**()**

Output for input values between 2,000 and 3,200, per problem’s requirement, and code execution results follow below:



# 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).***

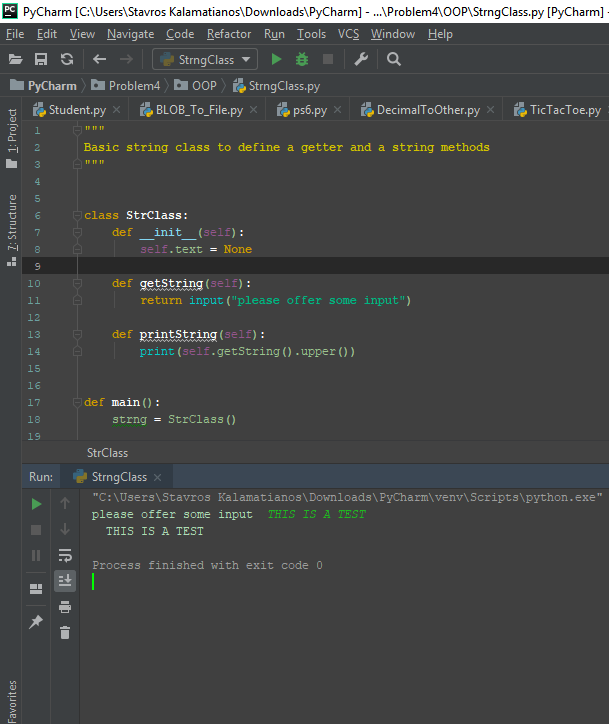
**Response to Problem 2:**

Python Source Code

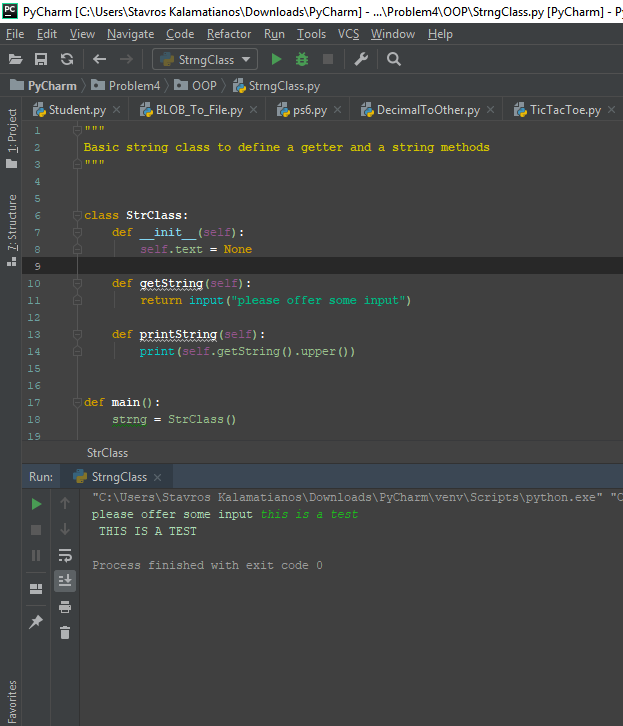
**"""  
Basic string class to define a user input getter and a print string method  
"""  
  
  
class StrClass:  
 def \_\_init\_\_(**self**):** self.text **= None  
  
 def getString(**self**):  
 return** input**("please offer some input")  
  
 def printString(**self**):** print**(**self.getString**()**.upper**())  
  
  
def main():** strng **=** StrClass**()** strng.printString**()  
  
  
if** \_\_name\_\_ **== "\_\_main\_\_":** main**()**

Input/Output execution results

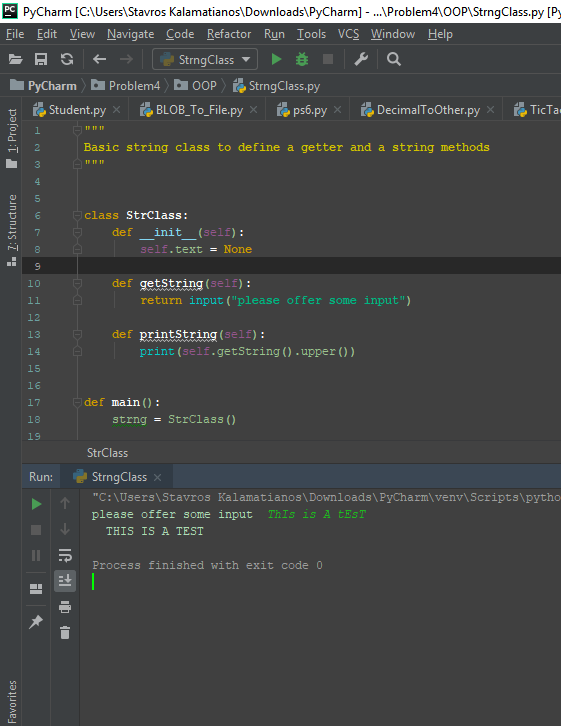
1. All Input is in CAPS



1. All Input is in lower case



1. Input is in Mixed mode, namely, in both CAPS and lower case





# 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.

**Response to Problem 3:**

R Source Code Start

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**# R Programming Exercise**

**# student: stavros kalamatianos**

**###################**

**#### Settings #####**

**###################**

**# do some memory cleanup.**

**gc(verbose = TRUE, reset = TRUE)**

**format(memory.size(), units = "MB")**

**options(scipen=999) # get rid of scientific notation**

**# reads the Renviron file to make this compatible with Rscript.exe**

**readRenviron("~/.Renviron")**

**# cleans out everything**

**rm(list=ls())**

**# defines a vector of courses**

**myCourses <- c('Physical Chemistry 500', 'Organic Chemistry 600', 'Transport Phenomena 550', 'Chemical Thermodynamics 456', 'Chemical Plant Design 610', 'Chemical Reactors I', 'Chemical Reaction Kinetics 570', 'Mass and Energy Balances 620', 'Analytical Chemistry II', 'Materials Science III')**

**# gets the length of the vector myCourses**

**length(myCourses)**

**# gets the first two courses from myCourses**

**myCourses[1:2]**

**# gets the 3rd and 4th courses from myCourses**

**myCourses[3]**

**myCourses[4]**

**# sorts myCourses using a method**

**sort(myCourses)**

**# sorts myCourse in the reverse direction**

**sort(myCourses, decreasing = TRUE)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

R Source Code End

Program Execution and output Start

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**> # R Programming Exercise**

**> # student: stavros kalamatianos**

**>**

**> ###################**

**> #### Settings #####**

**> ###################**

**>**

**> # do some memory cleanup.**

**> gc(verbose = TRUE, reset = TRUE)**

**Garbage collection 22 = 16+1+5 (level 2) ...**

**30.7 Mbytes of cons cells used (44%)**

**11.3 Mbytes of vectors used (18%)**

**used (Mb) gc trigger (Mb) max used (Mb)**

**Ncells 574341 30.7 1300254 69.5 574341 30.7**

**Vcells 1472046 11.3 8388608 64.0 1472046 11.3**

**> format(memory.size(), units = "MB")**

**[1] "58.63"**

**>**

**> options(scipen=999) # get rid of scientific notation**

**>**

**> # reads the Renviron file to make this compatible with Rscript.exe**

**> readRenviron("~/.Renviron")**

**>**

**> # cleans out everything**

**> rm(list=ls())**

**>**

**> # defines a vector of courses**

**> myCourses <- c('Physical Chemistry 500', 'Organic Chemistry 600', 'Transport Phenomena 550', 'Chemical Thermodynamics 456', 'Chemical Plant Design 610', 'Chemical Reactors I', 'Chemical Reaction Kinetics 570', 'Mass and Energy Balances 620', 'Analytical Chemistry II', 'Materials Science III')**

**>**

**> # gets the length of the vector myCourses**

**> length(myCourses)**

**[1] 10**

**>**

**> # gets the first two courses from myCourses**

**> myCourses[1:2]**

**[1] "Physical Chemistry 500" "Organic Chemistry 600"**

**>**

**> # gets the 3rd and 4th courses from myCourses**

**> myCourses[3]**

**[1] "Transport Phenomena 550"**

**> myCourses[4]**

**[1] "Chemical Thermodynamics 456"**

**>**

**> # sorts myCourses using a method**

**> sort(myCourses)**

**[1] "Analytical Chemistry II" "Chemical Plant Design 610" "Chemical Reaction Kinetics 570"**

**[4] "Chemical Reactors I" "Chemical Thermodynamics 456" "Mass and Energy Balances 620"**

**[7] "Materials Science III" "Organic Chemistry 600" "Physical Chemistry 500"**

**[10] "Transport Phenomena 550"**

**>**

**> # sorts myCourse in the reverse direction**

**> sort(myCourses, decreasing = TRUE)**

**[1] "Transport Phenomena 550" "Physical Chemistry 500" "Organic Chemistry 600"**

**[4] "Materials Science III" "Mass and Energy Balances 620" "Chemical Thermodynamics 456"**

**[7] "Chemical Reactors I" "Chemical Reaction Kinetics 570" "Chemical Plant Design 610"**

**[10] "Analytical Chemistry II"**

**>**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Execution and output End



# 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 a specific example of Principal Component Analysis(15 points)**

**Response to Problem 4:**

**2)**

The extraction of a combination of features from a data set [e.g., data compression] or a class of objects [e.g., human face pattern recognition] that describes it accurately enough with less information, that it is also insensitive to the variations of the class or set, is called feature extraction.

Principal Component Analysis [PCA] is a well-known and widely used feature extraction method.

Mathematically speaking, PCA is, most often than not, used for dimensionality reduction as a data processing technique which maps a high-dimensional space to a lower one with minimum loss of information.

**1)**

PCA aims to identify the principal components [PC] and their directions in a multi-dimensional space where they have the greatest variance that explains most of the information hidden in the data set.

The principal components correspond to the eigenvectors associated with the largest eigenvalues of the autocorrelation matrix of the data vectors.

In contrast, the principal components that correspond to the eigenvectors associated with the smallest eigenvalues of the autocorrelation matrix of the data vectors are called the minor components and they represent the noise in the data.

**3)**

The electromagnetic spectrum ranges from radio waves to gamma rays with only a small range of visible light detected by the human eye.

Hyperspectral sensors are electronic devices build to acquire a set of images from hundreds of narrow and contiguous bands of the electromagnetic spectrum from visible to infrared regions.

The capturing results then get combined to build a 3-D composite image. These images are called hyperspectral cubes and in geophysical, mineralogical, or metallurgical studies constitute a valuable tool for material identification, detection, and analysis of areas rich in minerals and metals, surface mapping or to facilitate the exploration of ores and petrochemicals.

Hyperspectral imaging is based on the premise that materials have different light reflection signatures based on the spectral band they usually react, thereby allowing the sensor to compose an accurate compositional cross-section image of any object.

For imaging to work efficiently similar image pixels must be grouped together. Pixels carry a big number of features by design. To classify patterns on images accurately the classification algorithm must be able to handle the interrelationship between sample sizes, number of pixel features required and its inherent algorithmic complexity.

Researchers have observed that the number of pixel features vs the number of training sets, necessary for classification, are correlated in an exponential fashion otherwise classification demonstrates poor performance. This phenomenon is coined the name “the curse of dimensionality”.

Increasing spectral resolution by dividing the spectrum used by hyperspectral imaging to a higher number of bands will not mitigate the above problem because of the strong correlation between adjacent bands.

There are 2 approaches to address dimensionality reduction. These are: feature selection and feature extraction.

Feature selection, as the name implies, is the process of selecting a subset of features that preserve crucial information, whereas feature extraction refers to the resultant transformed set of original features that will not jeopardize the accuracy of classification or clustering, nor will it increase its computational cost.

In Hyperspectral Imaging PCA is used in feature extraction. It initially, calculates the covariance

matrix of the given data set, and then finds the eigenvalues and eigenvectors of this matrix. Next, it selects a few eigenvectors of higher eigenvalues to form the matrix to transform the original data to a lower dimensional space.

As a result, the transformed data has its first few principal components contain almost all the original data variance, and the components in the new, information rich, feature space are uncorrelated in nature.

# 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).

**Response to Problem 5:**

5a)

Multiple linear regression and multiple logistical regression are both parametric regressions, they both utilize a linear equation to make predictions and they both fall under the supervised learning Machine Learning type of algorithms. But that is where their similarities stop.

Multi-Regression vs Logistic Regression Table of differences

The following table will attempt to list their differences that come to be quite a lot.

|  |  |  |
| --- | --- | --- |
| Difference | [Multiple] Linear Regression | [Multiple] Logistic Regression |
| Number of explanatory or independent variables | One for simple linear regression and more than one for Multiple Linear regression | Simple [Multiple] logit, or logistic, regression analysis is the regression of one binary, or dichotomous, categorical variable and one [Multiple] independent variable(s). |
| Dependent variable outcome | Continuous, not confined within a range | Discrete with 2 outcomes for binary logistic regression, e.g., True or False, 1 or 0, Yes or No  OR  multiple outcomes [nominal for example] for multinomial logistic regression  *Multinomial Regression Example:*  travel to NYC by car, bus, train, or plane  or  promotional level, e.g., Associate, Principal, Vice President, Director |
| Expectations [Result] | A linear relationship between dependent and independent variable(s) | No linear relationship |
| Purpose | Works on a variety of regression problems | Mostly used for Classification |
| Type of Model | Linear [straight line fit the data] | Non-Linear [fit a curve to the data] |
| Model’s deliverable | A dependent variable’s relationship with one or more independent variables | The probability of a binary outcome, multi-value nominal or ordinal outcome |
| Geometry | Straight line | S-shaped curve |
| Independent Variables Correlation | Although correlation is possible, the assumption that no major correlation is present holds | Must not be correlated [to avoid multicollinearity] |
| Method of Model Parameter Estimation | Least Square Estimation that calculates the coefficients that minimize the sum of squared distances of observed responses to fitted values | Maximum Likelihood Estimation that calculates the coefficients that maximize the probability of dependent Y to given X independent variable |
| Distribution Assumption | Dependent variable obeys a normal distribution | Dependent variable obeys a binomial distribution with a binary outcome |
| Error terms [ Residual] Normality | Error terms should be normally distributed | Error terms not expected to be normally distributed |
| Homoscedasticity | The randomness or noise in the relationship of each pair of dependent-independent variable, also expressed mathematically as an error term, is approximately the same across all pairs. | No need for residuals or error terms to be equal for each independent variable |
| Coefficient interpretation | Increase/decrease of dependent variable when Xi independent variable changes by 1 unit and the rest of n-1 independent variables remain constant, where i =1, …., n | Predicted odds ratio change when Xi independent variable changes by 1 unit and the rest of n-1 independent variables remain constant, where i =1, …., n |
| Goodness-of-fit-test | ANOVA | Hosmer & Lemeshow Test |
| Example | Predicting one’s salary based upon the years of experience, the company’s stock earnings per share and the number of projects he has been involved so far | Estimate odds ratio whether a student will get accepted to a specific University or not |

A proposed regression decision framework

The norm

We want to believe that the above table provides a detailed list of fundamental differences between multiple regression and logistic regression. The list can also be used to shed more light on which model to choose based on the nature of the dependent and independent variables of the model.

In the case of a binary or nominal dependent variable a logistical model could be a better choice whereas when the outcome has a continuous numerical value then multiple linear regression is a more suitable choice, especially if the number of exploratory variables is continuous numerical, more the one or even categorical that can be transformed to numerical by assigning numbers to different categorical classes.

A radical alternative

However, many practitioners and occasionally even academics, sometimes, recommend using multiple regression even if the dependent variable is a categorical one. If the result of that variable is above 1 then the decision would be to choose the category that responds to 1 and the reverse if the outcome ends up lower than 0.

Next, we will communicate a framework of model design that may potentially offer more flexibility to decide on a regression methodology when the dependent variable is categorical, beyond simply rely on its type.

*Decide based on multiple Data Sets*

We believe that a more thorough determining methodology would involve comparing a Linear vs a Logistic model with two or more copies of data sets. Each data set could describe the overall picture the model is trying to capture from a different perspective therefore, if, in the end, the same best model performer persists then we would have one more confirmation from the data which model does a better predictive job.

For example, a retail chain that evaluates the business decision to open more stores nationwide using different cuts of the data can test their regression models with data from different time periods or from different geographical locations.

Depending upon the derived results by each model [Multiple linear regression or (Multiple) logistic] the scientist can continue his analysis or stop, if, for example, both models assign very similar probabilities to the calculated output variable.

*Reselect Model based on Dependent Variables*

If there is a need to continue challenging the best choice, or look for a model with more predictive power, the next step could be to change the variables or their [linear or non-linear] combinations in the models, i.e., experiment with different regression models.

The data scientist can mix categorical and quantitative variables by, let us say, multiplying them or by creating other composite predictor variables, with exponents or logarithms [based on some logic or business rule] before he rebuilds new models of both kinds and then run another evaluation. In this case the independent variables of the logistical model will not be the standalone Xi variables anymore.

Instead, they could be Xi \* Xi+2, Xi+3 ^ Xi+5, or any other linear/non-linear combination, while the new independent variables will not disturb the linear nature of the final model.

A general regression equation to illustrate that can be:

Y = b0 + b1 \* X1 + b2 \* X2 +…. + bi \* (Xi \* Xi+2) + bi +1 \* Xi+1 + bi +3 \* ( Xi+3 ^ Xi+5) + bi +4 \* Xi+4 + bi +6 \* Xi+6 + …. + bn \* Xn

*Decide by splitting one Data Set*

Lastly, the assembled dependent/independent variables data used for modeling can be split in many ways to create a diverse collection of models and then compare them against each other. For example, the scientist can create models using only 75% of the data, by selecting a different 75% cut of the data before calculating the predicted vs real data errors searching for a better model.

Regardless of any aforementioned approach, it would not be prudent to compare the goodness-of-fit test for the two models because the models are different from each other, in the sense they do not predict the same type of variable, they serve different purpose, and they are constructed based upon different statistical assumptions.

5b)

# In this section, we used RapidMiner analysis environment to develop a Logistical model out of Heart Disease data found at <https://archive.ics.uci.edu/ml/datasets/Heart+Disease> and a Multi-Linear Regression model from student performance data delivered by Portuguese education taken from <https://archive.ics.uci.edu/ml/datasets/Student+Performance>

# The differences between the 2 types of analyses and their model features are listed below:

# Predicted Output

# Linear Regression used final grade field “G3” [in a scale of 0-20, where 20 is the highest grade] as the continuous dependent variable of a model developed to detect its linear relationship with the student’s family status by selecting a subset of family variables from the greater data set, whereas the predicted output for the Logistical model was the "goal" field that reflects the presence of heart disease (values 1,2,3,4) in the patient or its absence (value 0).

# Regression Problem

# Linear Regression can be used to predict a student’s grade based on his family status, including parent’s job, parent’s marital status, parent’s education, and family size.

# On the other hand, the Logistical model can predict the presence of heart disease in a patient from his age, gender, and a series of measurements of medical parameters such as cholesterol, sugar levels, current psychological status, for instance depression and quality of life habits like regular exercising.

# Model focus

# The goal of the Linear Regression is to find the best fit line that will enable the most accurate output prediction.

# 

# Fig. 1 Table of Linear Regression statistics of modeling attempt 1

# Fig. 1 lays out the parameters of the linear model and among other statistics their p-values showing greater than the typical threshold p=0.05 that, in turn, increases our confidence on the null hypothesis stating the linear regression coefficients = 0.

# Attempt’s 1 Root Mean Square Error = 1.80 and correlation coefficient R2 = 0.851

# Removing all independent variables except G2 returns a model with statistics closely valued to the above ones [see Fig. 1a] with Root Mean Square Error = 1.906 and correlation coefficient R2 = 0.849

# 

# Fig. 1a Table of Linear Regression statistics of modeling attempt 2

# The Logistical Regression tries to create the S-curve graph that classifies cases between the two classes, in this case, True [the patient suffers from Heart Disease] or False [Heart Disease is not present].

# Logistic Regression’s S-curve models the probability for classification of a patient having Heart Disease or not. In the next 2 RapidMiner visualizations we provide the positive or negative magnitude of change of the estimated odds for a patient to have Heart Disease vs. not to have Heart Disease, when any of the independent variables change by a unit of measurement.

# In Fig. 2 the exponential of the weight attributes of the logistical model show the amount of change in the estimated odds of the dependent variable happening than not happening for each feature [independent variable] when the rest of the features remain unchanged. The interpretation changes whether the feature is a categorical or a numerical variable.

# For example, for a unit increase of numerical feature thalach [patient achieved a higher-than-average maximum heart rate] the odds of Heart Disease vs not, decreases by exp(-0.19) = 0.827 when the rest of the features remain constant, which makes perfect sense because healthy people tend to exhibit a quite higher maximum heart rate (around 160) compared to the maximum heart rate (less than 150) of patients with a heart disease.

# For patients with high values of continuous feature ca [number of major vessels (0-3) colored by flourosopy] the odds of having heart disease vs not having it, increases by exp(1.01) = 2.745 times for one unit increase of the ca value when the rest of the features remain the same.

# Again, a higher value of ca denotes that major blood vessels that supply oxygen and nutrients to the heart are damaged, becoming rigid and narrow resulting in inhibiting the blood flow that creates heart problems.

# Lastly, for patients with exang [exercise induced angina] the odds of having heart disease than not, are by a factor of exp(0.25) = 1.28 higher than patients who do not exhibit exang, an intuitive consequence if exercise reduces blood flow to the heart.

# 

# Fig. 2 Logistic Regression model weights for all independent variables used in the model

# Fig. 3 is the confusion matrix of the logistic model. The number of True Positive [TP] cases predicted were 47 with 7 False Positives [FP], the True Negatives [TN] were 29 with 8 False Negatives [FN].

# Fig. 3 Confusion matrix for the 2-class classification

# The overall correctly positive cases predicted define the Recall Rate [PR] = TP / (TP + FN)

# The number of positive cases predicted out of all positive cases describes the model’s Precision Rate = TP / (TP + FP)

# Finally, the Accuracy = (TP + TN) / Total = 76 / 91 = 83.52% is the measure of the number of cases the model predicted correctly overall.

# Based on the evaluation results from its confusion matrix, the logistic model shows quite robust with good signs of predictability worth relying.

# Method of parameter estimation

# We will recall the error measure of the Linear Regression model, that is reported as the Root Mean Squared Error [RMSE] from linear regression attempt 2, to be equal to 1.906, whereas the cross entropy of the Logistical model was 0.585.

# Although in the Linear-Logistic differences table we reported Maximum Likelihood Estimation as the method of parameter estimation of the Logistic model, here will rely on cross-entropy, RapidMiner’s criterion of performance, that practically can also yield a satisfactory model composition with an equally good parameter estimate.

# Output Data Type

# The Linear Regression model is built for a student’s grade estimation though the Heart Disease Logistic model to identify whether a patient needs treatment because of a cardiovascular disease condition.

# In other words, in the former model case the result will be a numerical student grade between 0 and 20, whereas in the latter it will be binary response, True or False.

# Dependent/independent feature relationship

# The Linear Regression model exhibits adequate statistical evidence of a satisfying linear relationship between dependent and independent variables with a Root Mean Square Error = 1.906 and a correlation coefficient R2 = 0.849

# Likewise, the Logistical model demonstrates its predictive ability via an accuracy of 83.52% and a confusion matrix with three out of four statistics calculated in class recall/precision ranging above 80%.

# 

# Collinearity in Linear Regression

# As discussed before, there can be collinearity among features in the case of Linear Regression but not in the case of Logistic Regression, because of the inherently different way Logistic Regression method functions.

# To detect collinearity RapidMiner supports the calculation of the squared correlation R2, that, conveniently, is a parameter of the Correlation Matrix Process Operator.

# The Correlation Matrix for all the family features picked for the Linear Regression model in attempt 1 is listed below. We picked attempt 1’s matrix to show because it’s a better demonstration of the software’s capability.

# As one can quickly observe there is hardly any feature pair that exhibits a significant correlation.

# 

# The same Correlation Matrix for the features of the Logistic model was also computed and it is presented below for the reader’s review. Again, no independent variable pair shows a worth mentioning correlation, satisfying the theoretical requisite of a Logistic model.

# 

# Conclusion

# In the last section of this report, we tried to briefly explain how Logistic and Linear regressions are different from each other by listing a series of characteristics they both exhibit in a dissimilar way.

# Then we downloaded two candidate data sets to develop a model of each regression to explore some of those differences for demonstration purposes. These sets were Heart Disease data with medical features relative to the condition, to identify whether a patient has it or not by using a Logistic model and a body of student family/educational performance data from Portugal to predict a student’s grade using a Linear Regression model based upon the independent variables that showed her/his family status.

# Both models exhibit strong signs of acceptable regression with low collinearity. Verbal explanations and several visual aids are presented as supporting evidence to help the reader understand the analysis results.

# This section of the report is an effort to demonstrate the student’s understanding of two very useful and indispensable statistical tools to anyone who works in the discipline of data analytics, namely, Linear and Logistic Regression.