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: Erick Orellana Morales GMU G#00697803

Student Signature (Honor Certification): *Erick Orellana Morales*

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**.

#Get input from user

user\_number = int(input(*'Please enter a number: '*)) #Number from user to check against requirements

#Constants used for calculations

divisible\_by = 7

multiple\_of = 5

min\_range = 2000

max\_range = 3200

#Variables to hold results of tests of program requirements

divisible\_by\_constant = False

multiple\_of\_constant = False

is\_in\_range = False

#Confirm if user input is a divisible by the given constant (Requirement A)

if ((user\_number % divisible\_by) == 0):

divisible\_by\_constant = True

else:

divisible\_by\_constant = False

#Confirm if user input is a multiple of the given constant (Requirement A)

if ((user\_number % multiple\_of) == 0):

multiple\_of\_constant = True

else:

multiple\_of\_constant = False

#Confirm if user input is within the provided range, exclusive of boundaries (Requirement B)

if (user\_number < max\_range) and (user\_number > min\_range):

is\_in\_range = True

else:

is\_in\_range = False

#Print results based on the requirements

if (divisible\_by\_constant == True) and (multiple\_of\_constant == False) and (is\_in\_range == True): #All requirements are met

print(*"\n"* + *"The number {} meets all the requirements!"*.format(user\_number))

else: #Determine which requirements were not met

print(*"\n"* + *"The number {} does not meet all the requirements!"*.format(user\_number))

print(*"\n"* + *"Here is an explanation of this result:"*)

if(divisible\_by\_constant == False):

print(*"- This number is not divisible by {}. (Does not meet requirement A)"*.format(divisible\_by))

if(multiple\_of\_constant == True):

print(*"- This number is a multiple of {}. (Does not meet requirement A)"*.format(multiple\_of))

if(is\_in\_range == False):

print(*"- This number does not fall between {} and {}. (Does not meet requirement B)"*.format(min\_range,max\_range))

Note: I interpreted the prompt to mean exclusive of the max and min boundaries on requirement “A” (noted in the code). To change the code to inclusive of the max and min boundaries, line 38 (view screenshot) would be updated to the following:

Exclusive (as currently noted in my code)

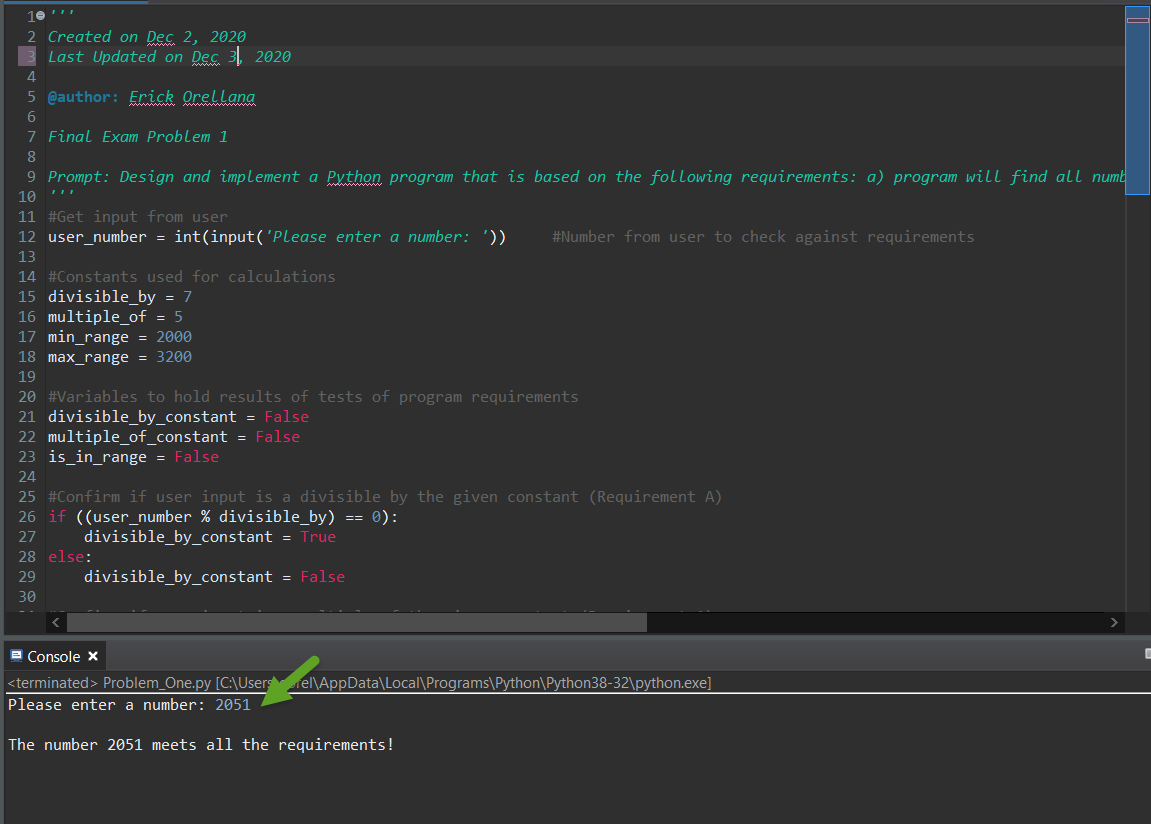
if (user\_number < max\_range) and (user\_number > min\_range):

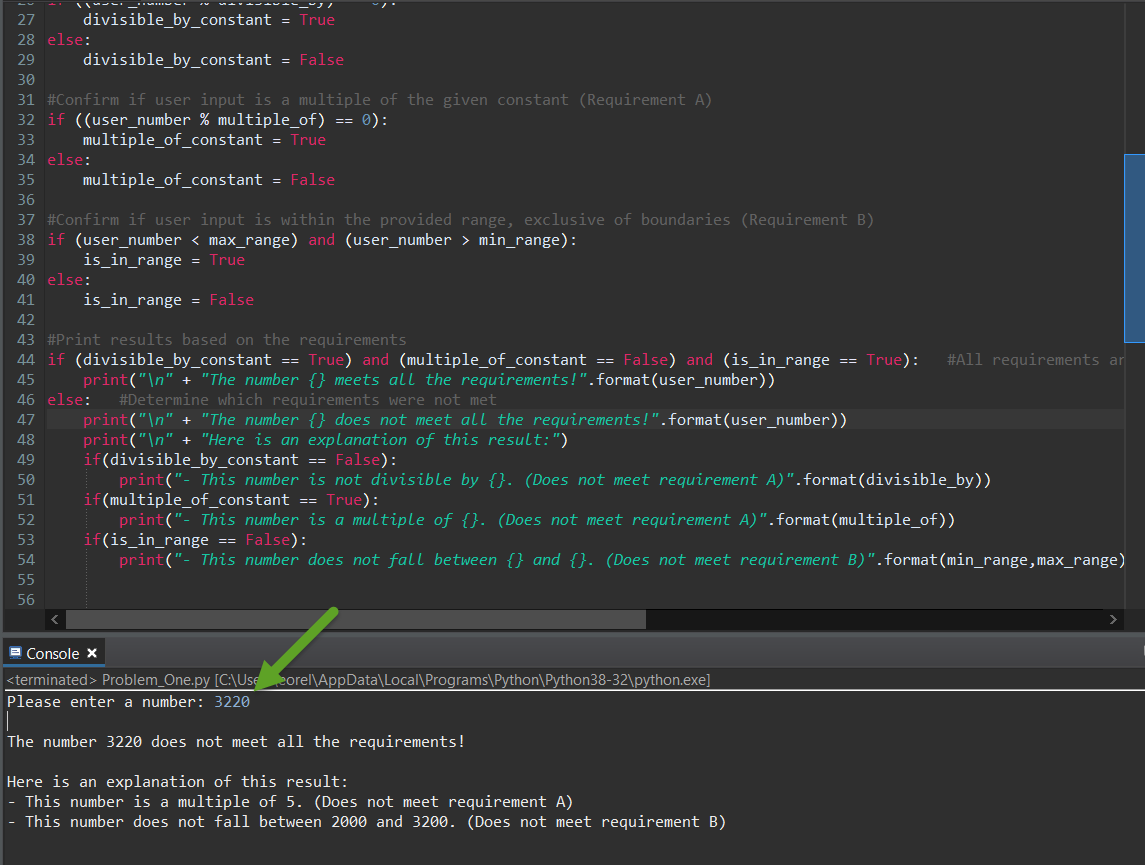
Inclusive (not in my code)

if (user\_number <= max\_range) and (user\_number >= min\_range):

See screenshots below (using Eclipse):

1. A number that meets all the requirements
2. A number that does not meet all the requirements





# 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).***

#Class to get a string from a user, convert the string to all upper case letters, and print the results

class **StringConversion**:

#Initiating variables

def **\_\_init\_\_**(*self*):

*self*.user\_input = *''*

#Get a string from the user

def **getString**(*self*):

*self*.user\_input = input(*"Enter a string to convert to upper case letters: "*)

#Print the string in all upper case letters

def **printString**(*self*):

print(*"\n"* + *"Here are your results -"*)

print(*"You entered the string: "* + *self*.user\_input)

print(*"In all upper case letters this string is: "* + *self*.user\_input.upper())

#Calling the class and methods to get a string from the user and show the string in all upper case letters

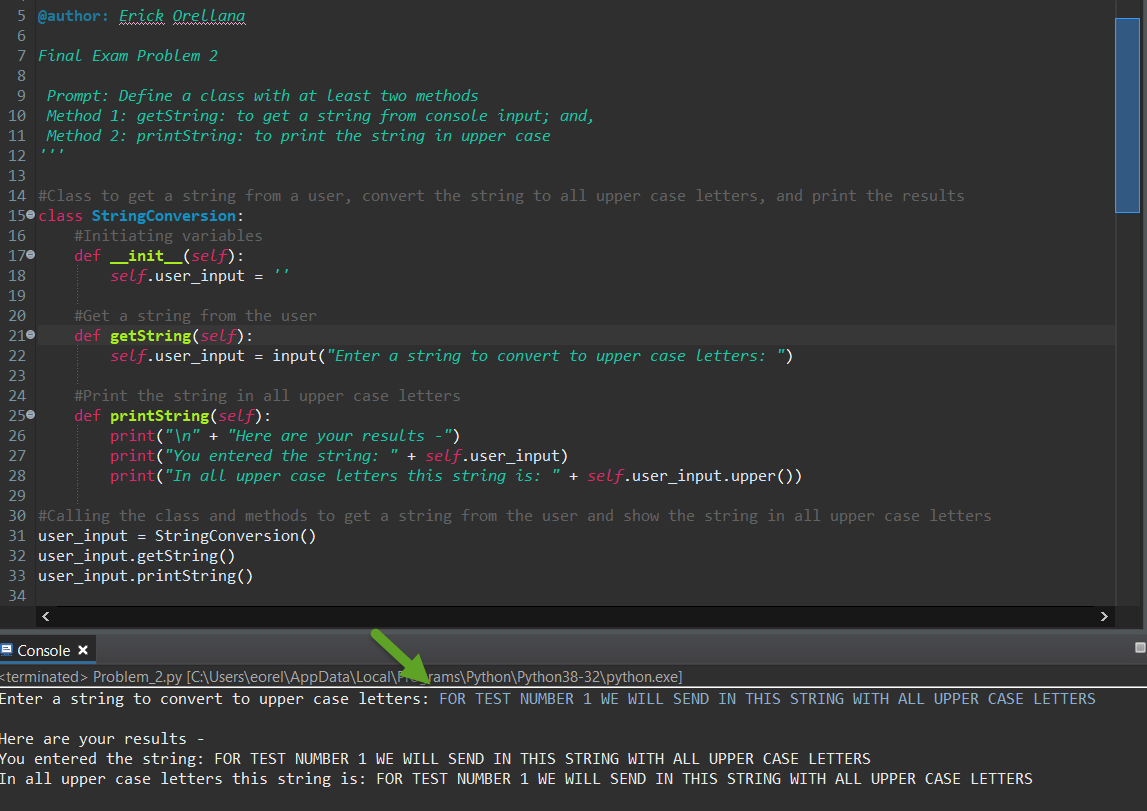
user\_input = StringConversion()

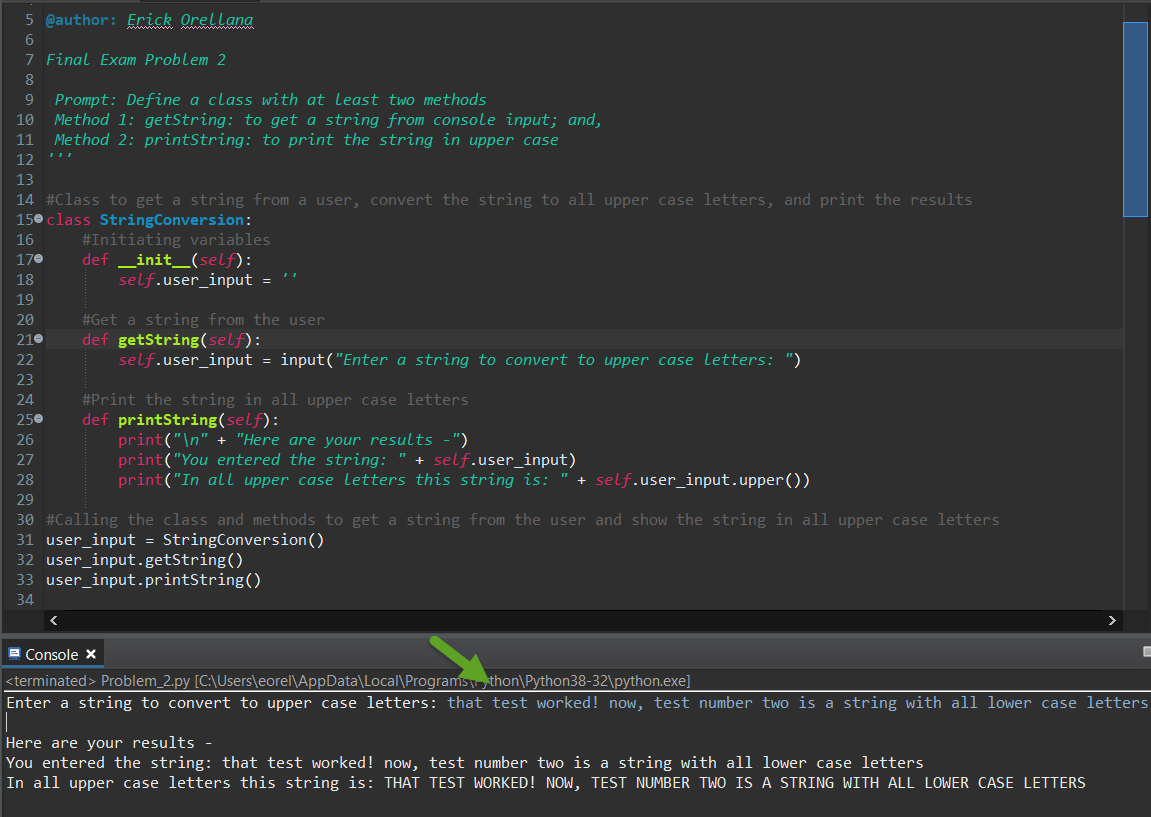
user\_input.getString()

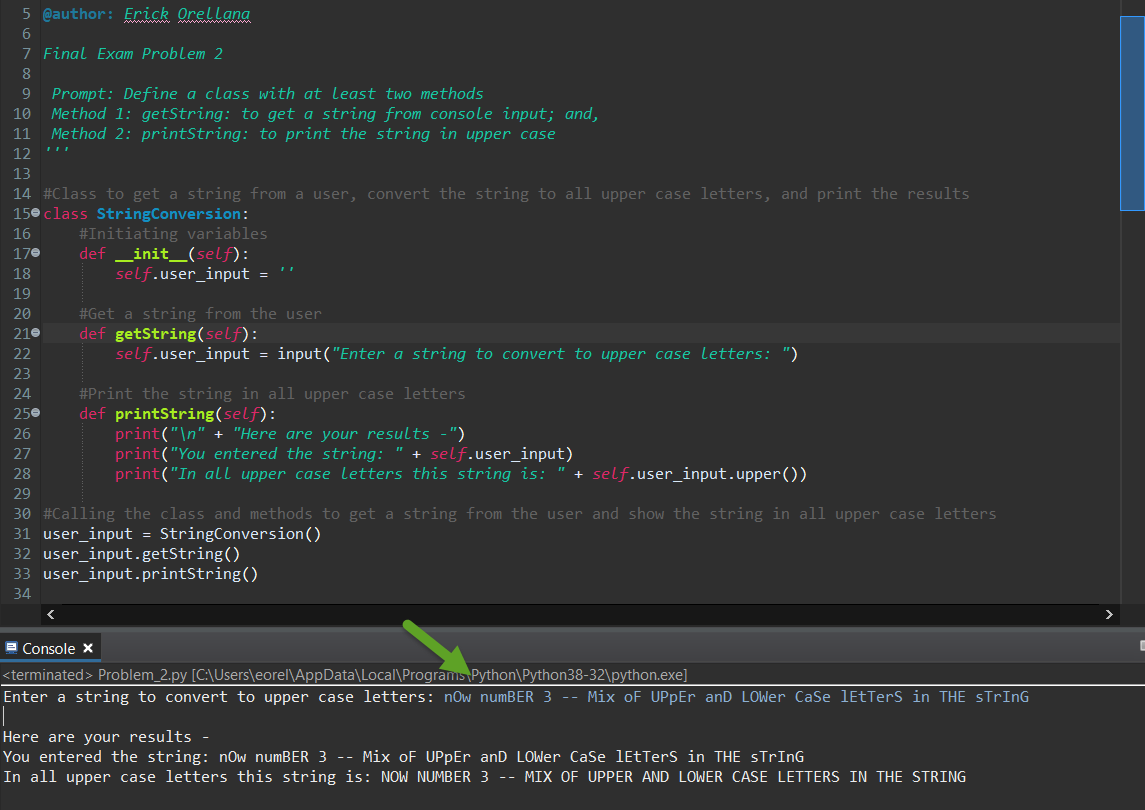
user\_input.printString()

See screenshots below (using Eclipse):

1. Input string: All upper case letters
2. Input string: All lower case letters
3. Input string: Mix of upper and lower case letters









# 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.

#Part 1: Vector of at least 8 courses

myCourses <- c("IT 106", "IT 206", "MATH 113", "CHEM 211", "ENGR 107", "STAT 344", "ARTH 201", "HAP 360", "ACCT 203", "MSOM 300")

#Part 2: Length of vector myCourses

length(myCourses)

#Part 3: The first 2 courses from myCourses

myCourses[1:2]

#Part 4: The third and fourth courses from myCourses

myCourses[3:4]

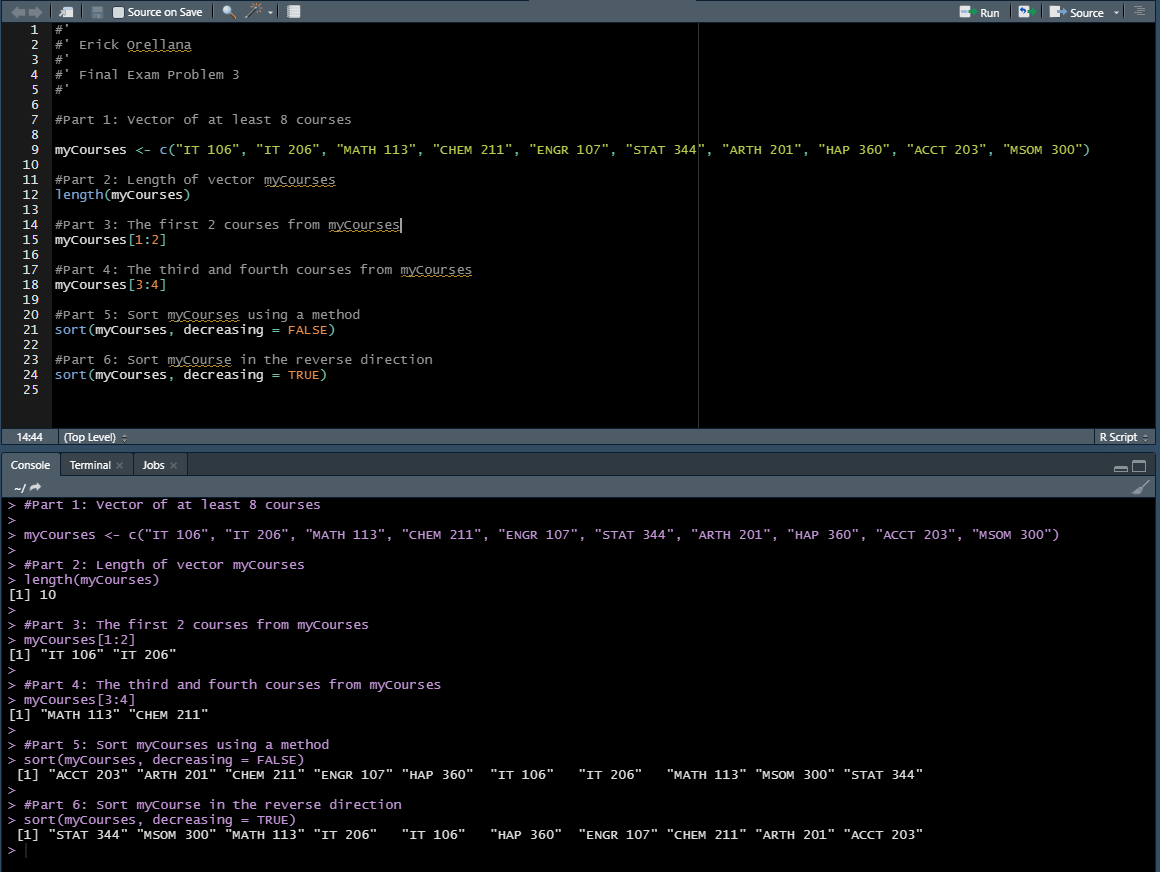
#Part 5: Sort myCourses using a method

sort(myCourses, decreasing = FALSE)

#Part 6: Sort myCourse in the reverse direction

sort(myCourses, decreasing = TRUE)

See screenshots below (using RStudio):





# 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)**

1) A component is a variable created by combining predictor variables from a data set. Components are intended to be an uncorrelated representation of the original predictor variables.

2) Principal Component Analysis is a process that looks at transforming predictor variables into a potentially smaller number of components to use for analysis while adhering to a high data variability standard. The variability standard is set on a case by case basis. The components generated from this analysis can then be used as variables in machine learning algorithms for example. The intent of a Principal Component Analysis is to simplify the successive analysis that will be completed by having components that have kept as much of the useful information from the original dataset while stripping out outlines, general noise, or redundancy in the data.

3) An example of Principle Component Analysis is the following:

A field management software company is experiencing issues with customer retention in the first 12 months of subscription. To combat this issue the company has collected the following information on accounts that have been paying members for over 12 months:

* Monthly rate paid for subscription
* Number of standard features used in the software
* Number of premium features used in the software
* Number of user licenses purchased for the account
* Average response time to requests submitted by customers to customer support
* Resolution rate on tickets submitted by customers to technical support
* Average number of hours that customers use the software per week

A Principal Component Analysis can be used to narrow down these different variables before continuing with further analysis. Ultimately this information can be used to improve retention by getting a better understanding of what makes a customer stay with the company past 12 months to build corporate strategies around it.

# 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).

1) The main difference between Multiple Regression and Logistic Regression is that Multiple Regression models a linear or continuous relationship between one or more predictor variables on a quantitative or continuous response variable while Logistic Regression models a probability based on one ore more predictor variables that results in a binary or categorical response variable.

A circumstance that would dictate the use of a Logistic Regression model would be if the analysis being done is being used to determine if the outcome either is or is not something. An example of this is would be creating a model to predict if an applicant will get an internship based on GPA. In this situation, the outcomes would be either that the student did get the internship (1) or the student did not get the internship (0). The predictor variable would be the student’s GPA.

A circumstance that would dictate the use of a Multiple Regression model is if a prediction needs to be made that would result in a quantitative or continuous response rather than a categorical response. An example of this would be creating a model that would show the relationship between salary based on years of experience in a field and years of post-secondary education. In this case the predictor variable would be the years in the field and/or the years of post-secondary education and the quantitative variable would be the salary.

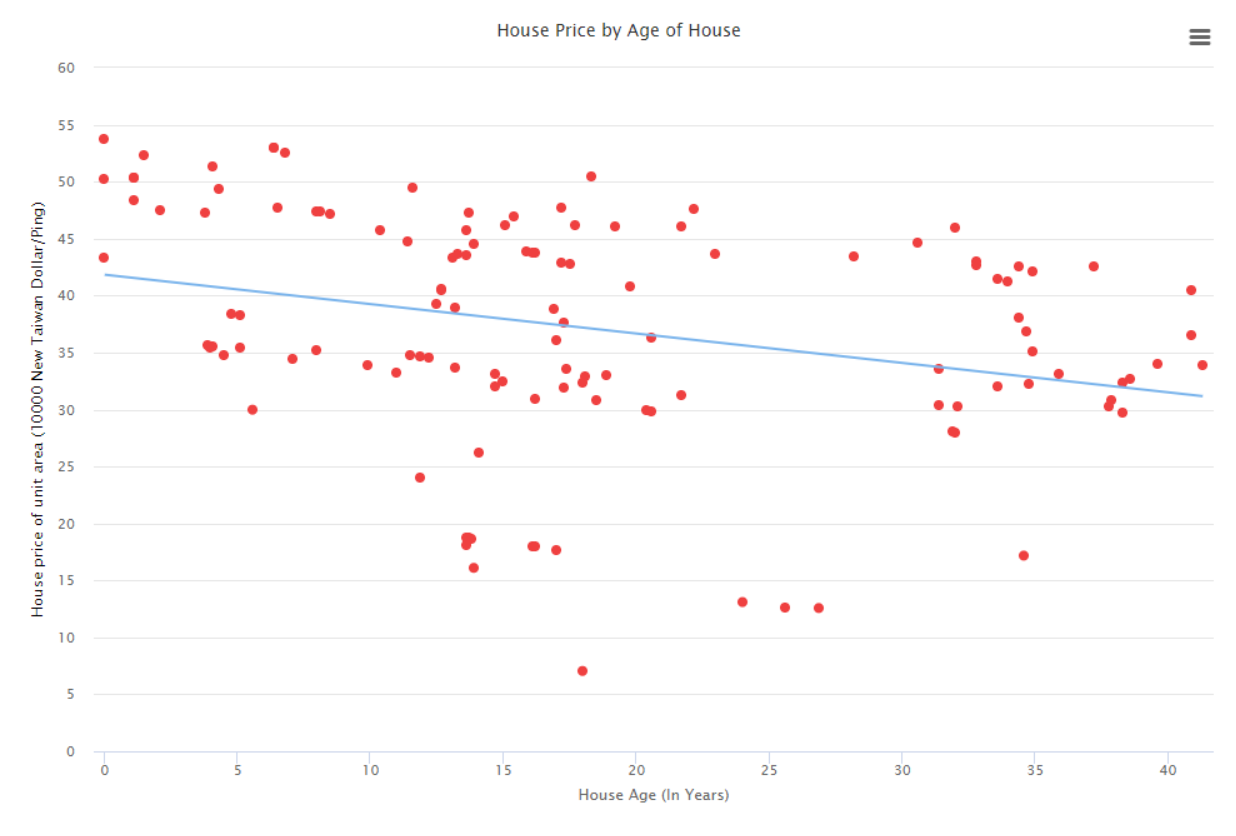
2)

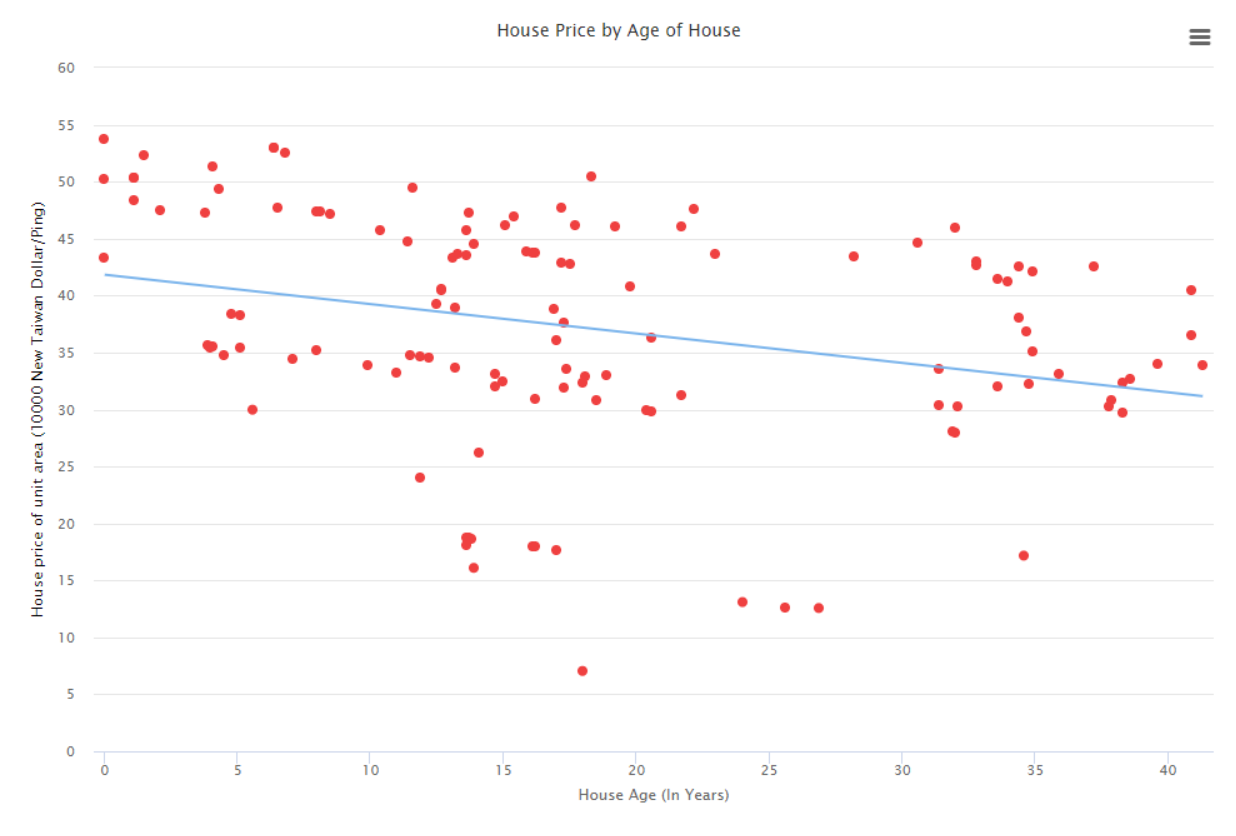
**Multilinear Regression**

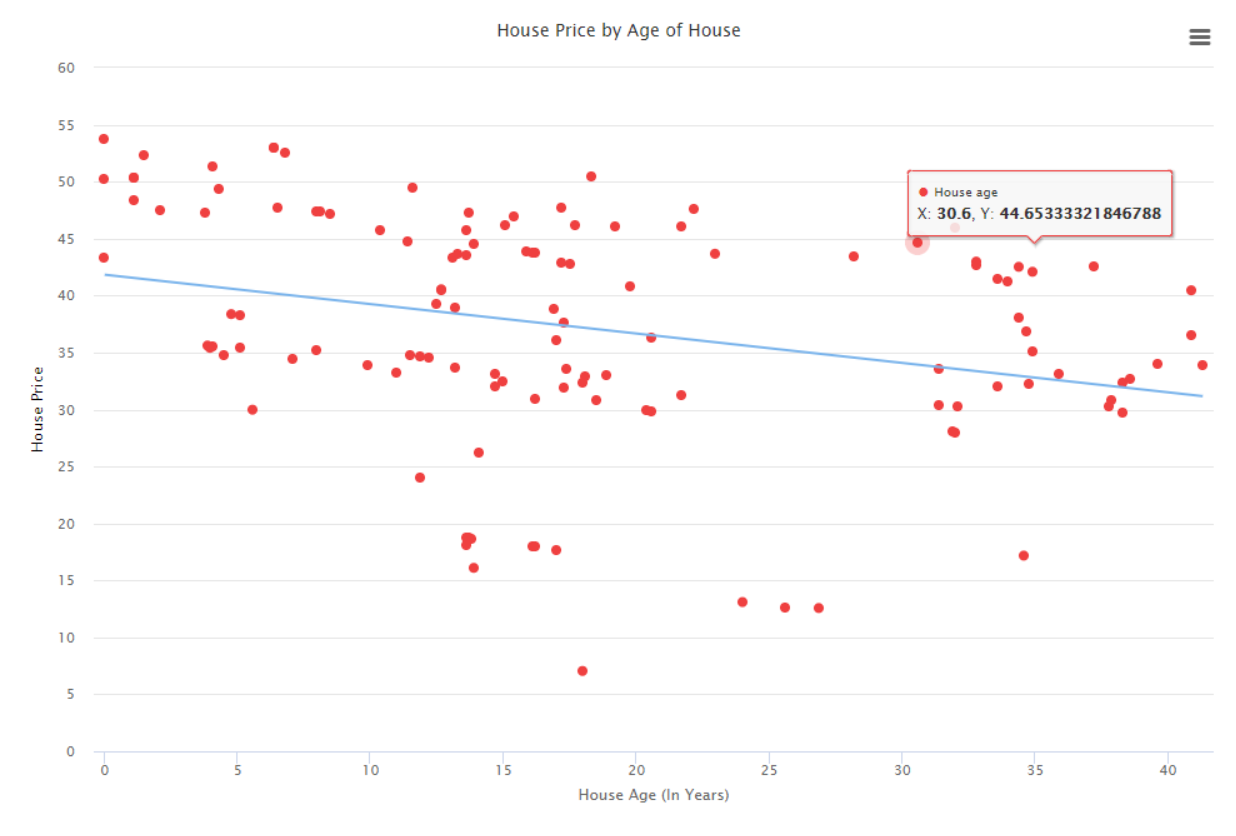
Data Used: <https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set>

Sample model: The impact of house age, distance to nearest MRT, and number of convenience stores within the living circle on house price

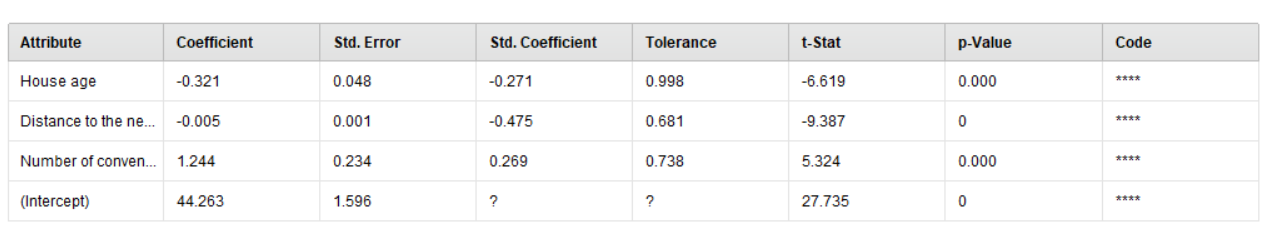
Graph (from RapidMiner)

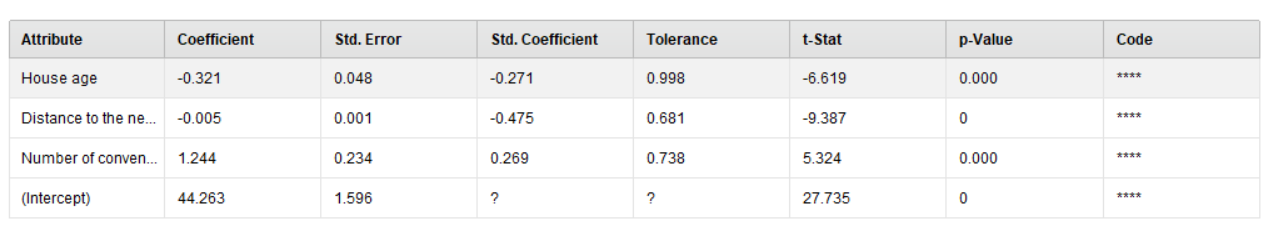


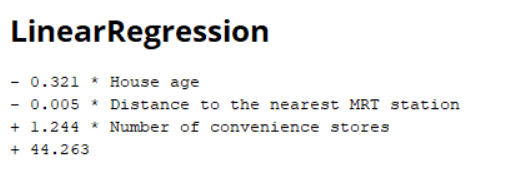


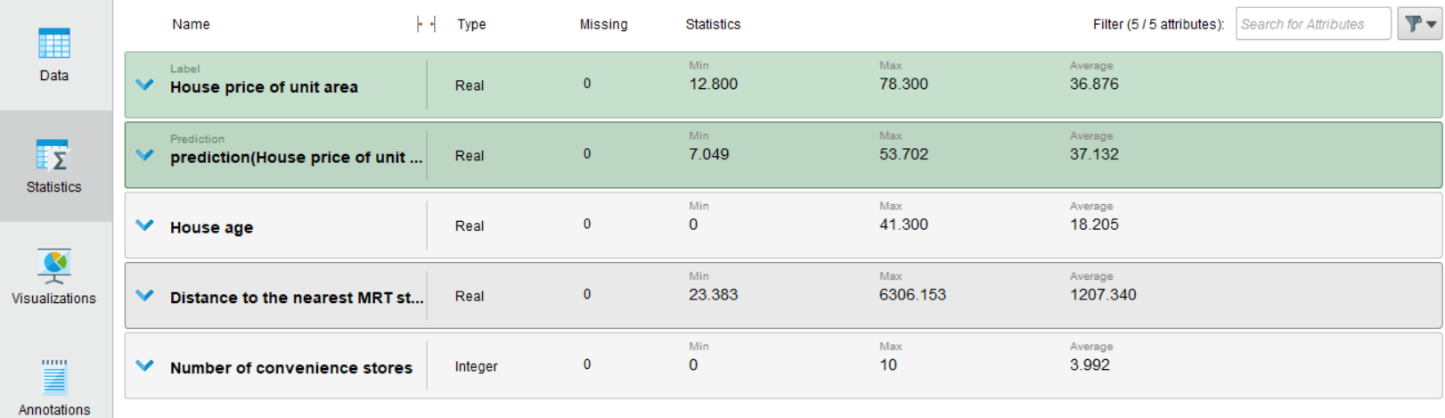


This model can be used to predict the price of a house based on the age of the house. The result is a quantitative/continuous value.

Additional Details (from RapidMiner)







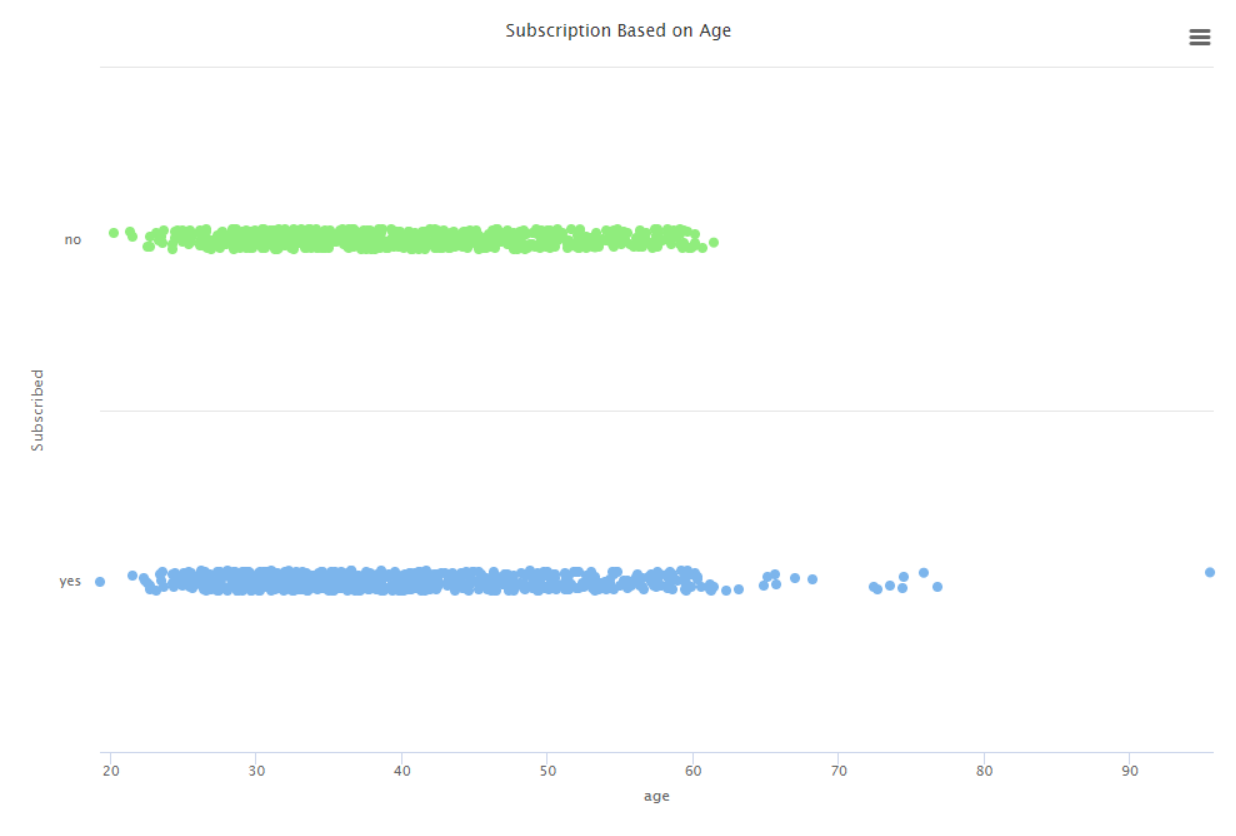
**Logistic Regression**

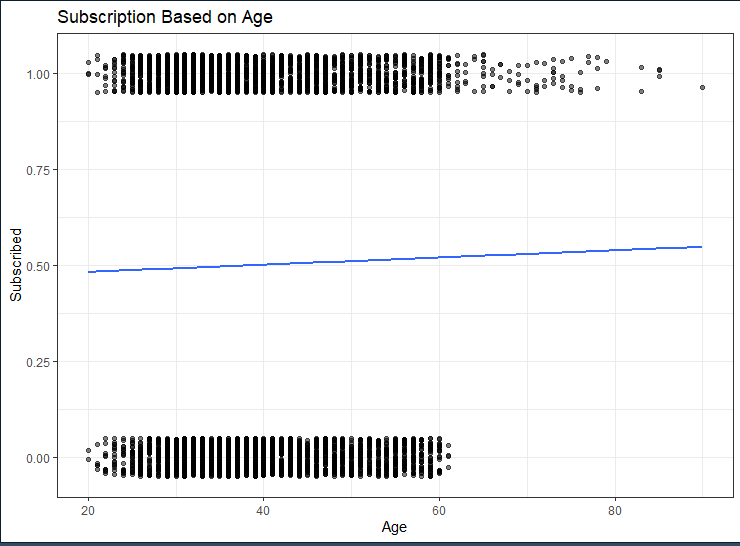
Data Used: <https://archive.ics.uci.edu/ml/datasets/Bank+Marketing>

Sample Model: Will a client subscribe to a term deposit based on client’s age and having a housing loan already.

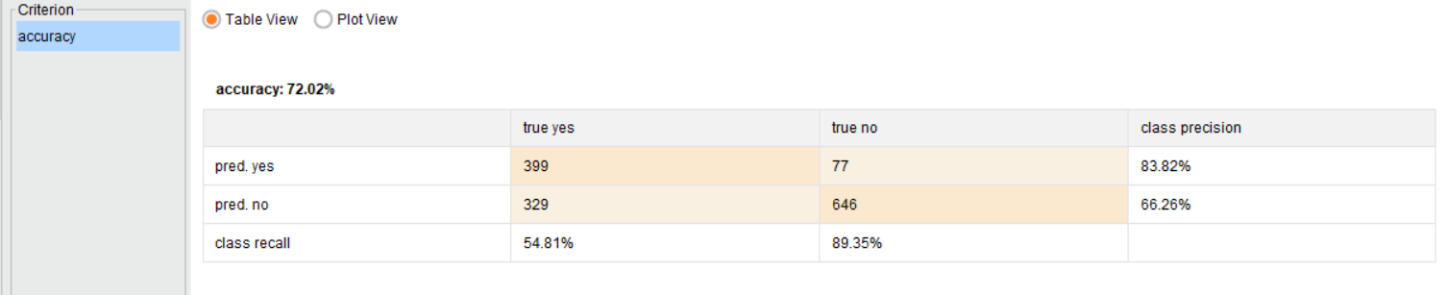
Graph (from RapidMiner and RStudio):

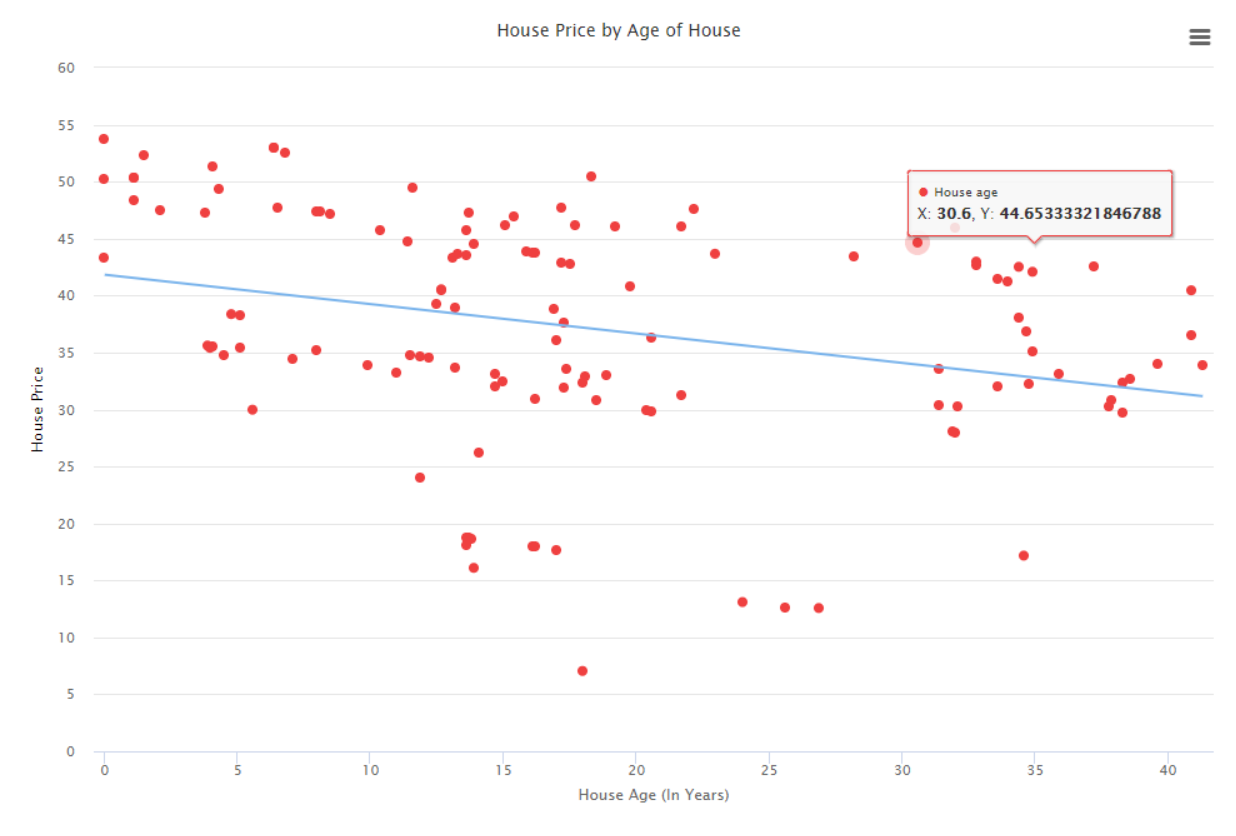
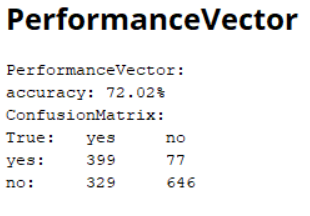
RapidMiner

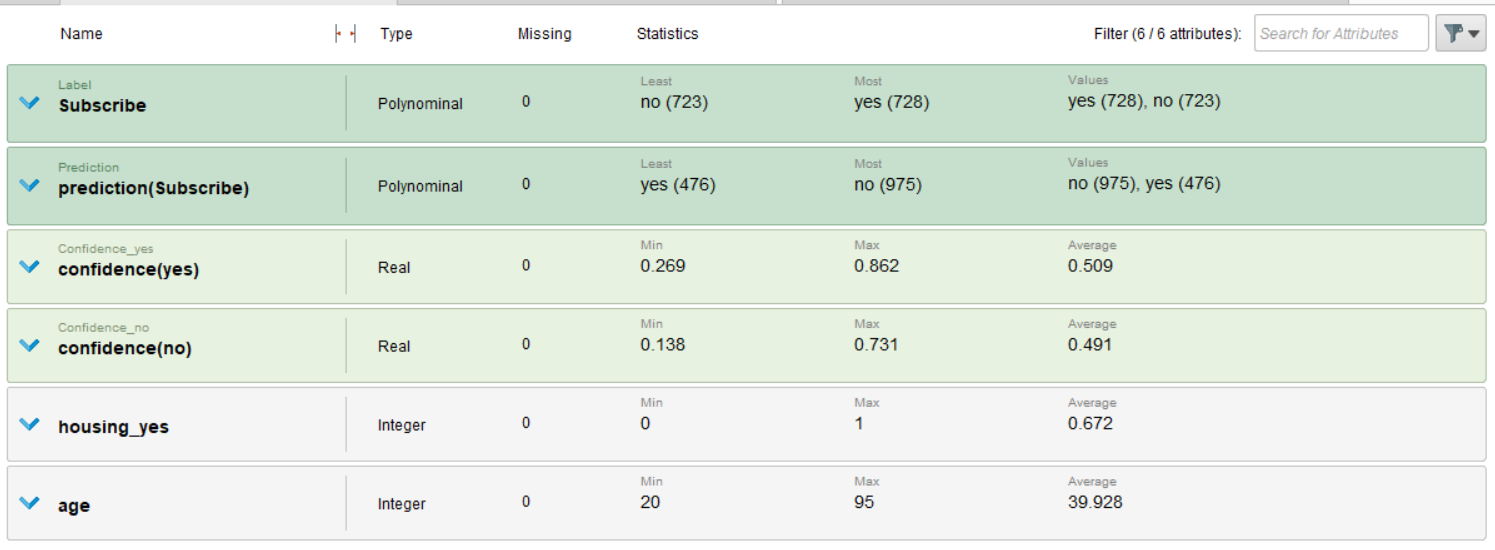


RStudio

This model can be used to predict if a client will or will not subscribe based on age. As shown here, the result/output is either yes they will subscribe (1) or no they will not subscribe (0). The trend line for logistic models can at times appear to have more of an “S” shape based on the data being entered into the model.

Additional Information (from RapidMiner):





*R Code for graph:*

geom\_jitter(width = 0, height = 0.05, alpha = 0.5) +

theme\_bw()

data\_space

data\_space <- ggplot(Loan\_Accepted2\_middle, aes(x = age, y = Subscribe)) +

ggtitle("Subscription Based on Age") + xlab("Age") + ylab("Subscribed") +

geom\_jitter(width = 0, height = 0.05, alpha = 0.5) +

theme\_bw()

data\_space

data\_space +

geom\_smooth(method = "glm", se = FALSE, method.args = list(family = "binomial"))

