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: Ashritha Chitimalla GMU G#: 00890574

Student Signature (Honor Certification): Ashritha Chitimalla

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**.

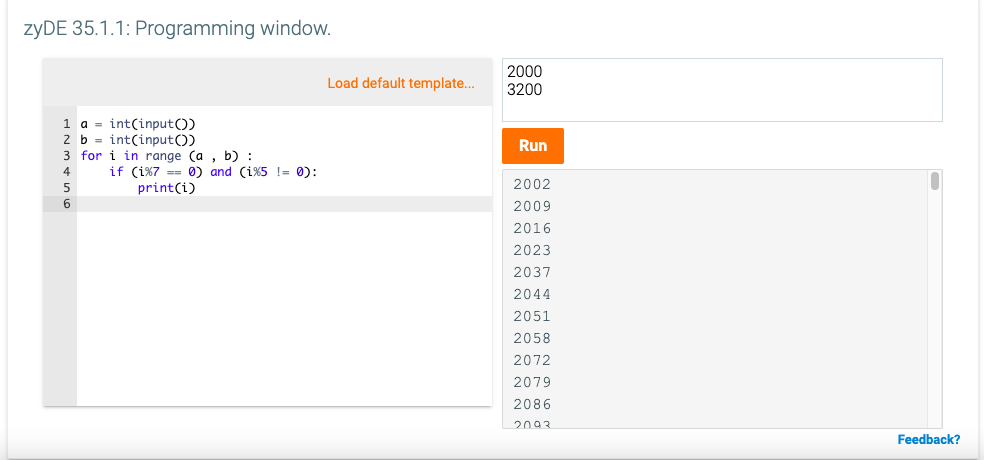
a = int(input())

b = int(input())

for i in range (a , b) :

if (i%7 == 0) and (i%5 != 0):

print(i)



# 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).***

def getString():

return input()

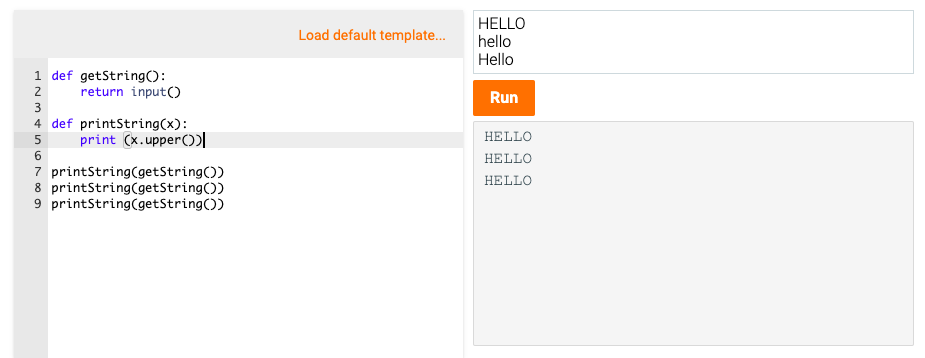
def printString(x):

print (x.upper())

printString(getString())

printString(getString())

printString(getString())





# 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.

**x <- c("MAT101", "CS110", "CS212", "ENG301", "STAT310", "CS450", "CS484", "DEAN500", "OR531")**

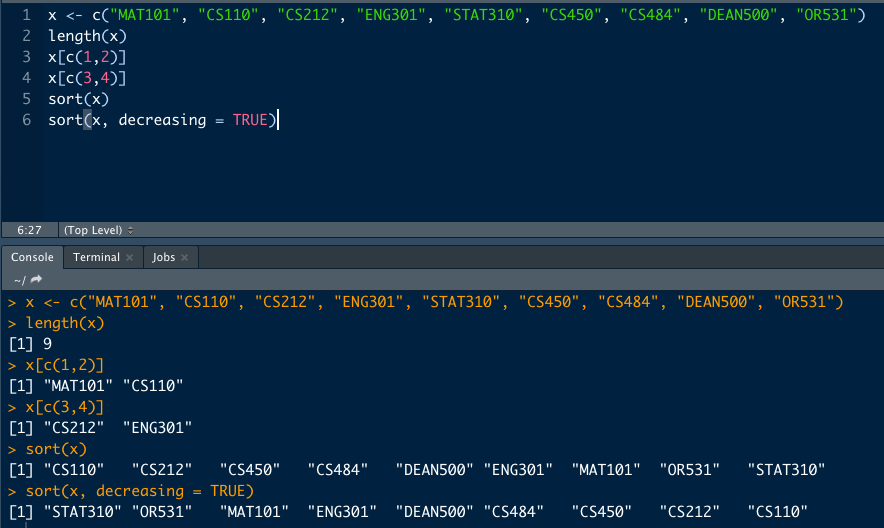
**length(x)**

**x[c(1,2)]**

**x[c(3,4)]**

**sort(x)**

**sort(x, 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)**
4. In a dataset, component refers to a variable that is a combination of several other highly correlated variables that are used as a predictor with each predictor having a weight in determining the best. An example I have learned in the course is the use of GPA and test scores to predict the academic success of the freshman. Let’s say, a given dataset includes GPA, test scores, sports, and extracurricular activities. GPA and tests scores are strongly correlated and contribute as a component and sports and extracurricular activities are combined to be called a component.
5. Principle Component Analysis is a statistical technique used for dimensionality reduction. PCA converts the variables that are given into components to simplify the analysis. The intention of PCA is to maximize the variability using as few components as possible.
6. Here is an example of PCA being used on the US census demographic dataset. The dataset I found comes with 37 attributes such as the number of White, Hispanic, and Black people, median household income, percentage of people under the poverty level, etc.

Having 37 attributes adds a lot of noise in the regression testing. Using PCA, I was able to reduce the number of features down to 5 of the best combinations of data. This kept the most significant bits of data while cleaning out the rest of the disturbance. Having fewer factors improved two factors for my predictions: 35% faster results and 20% accuracy improvement.

# 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.

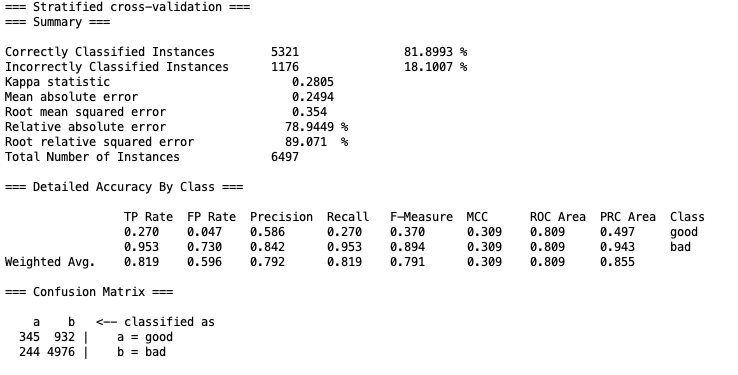
Step 1: Perform a quick search of the [UCIS public data archive](https://archive.ics.uci.edu/), a well-curated site which you already have seen as part of your introductory RapidMiner training.

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

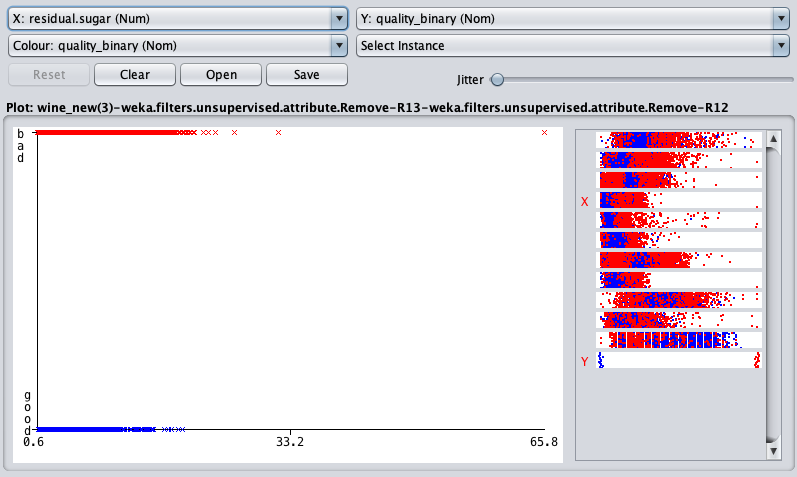
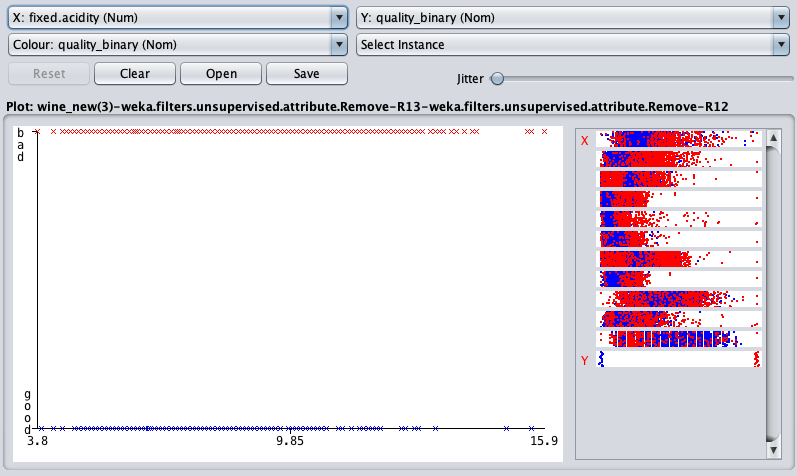
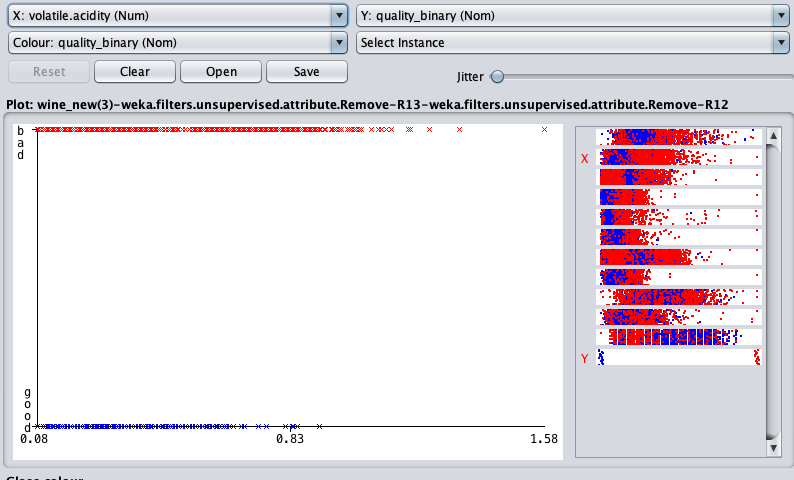
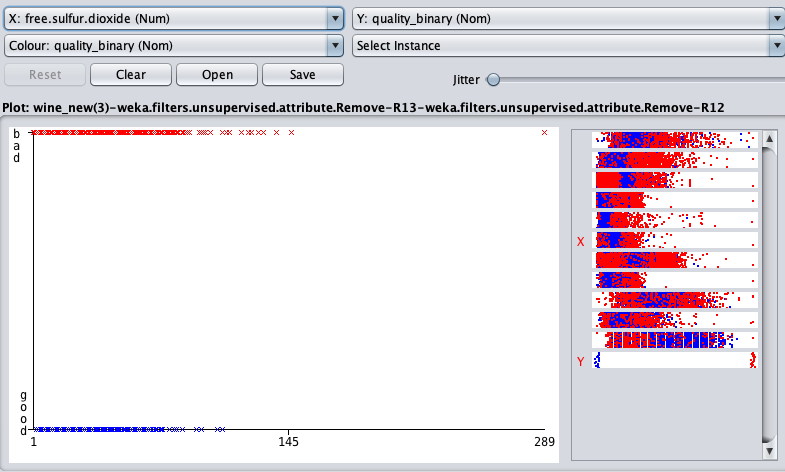
Step 3: Run regression, .and use visualizations to demonstrate the conceptual answers you provided for 5.(a).

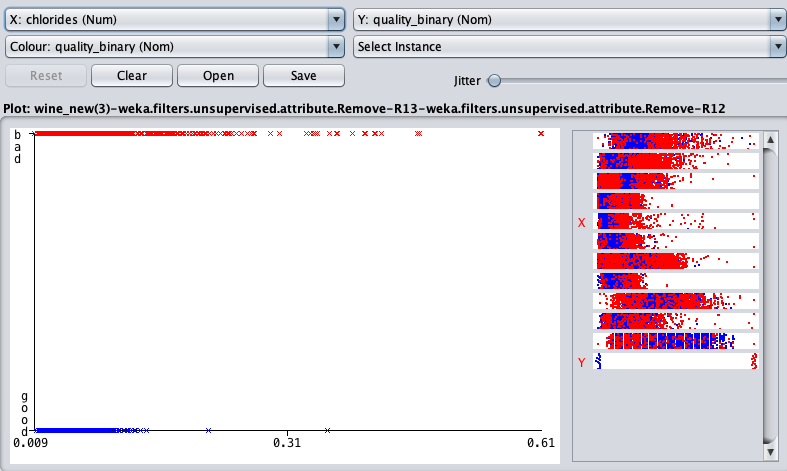
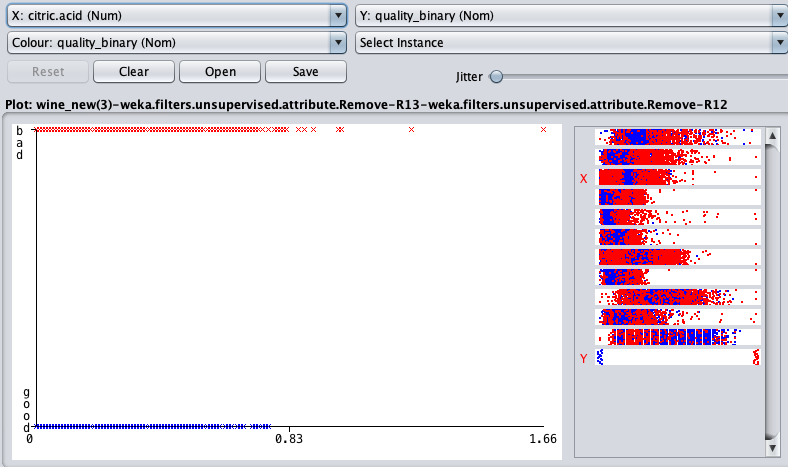
1. Multiple Regression is an extension of simple linear regression where one variable is linearly mapped to another variable. In multiple regression, we map multiple independent variables to a single dependent variable. Logistic regression is used for classification. It uses a function to determine the value of a binary dependent variable. You would have different use cases for the two functions based on what variable you are looking for. If you are looking for a binary value, for example yes/no, alive/dead, male/female, you would want to use logistic regression. however, when looking for a value on a range, you would want to use multiple regression.
2. Logistic Regression:

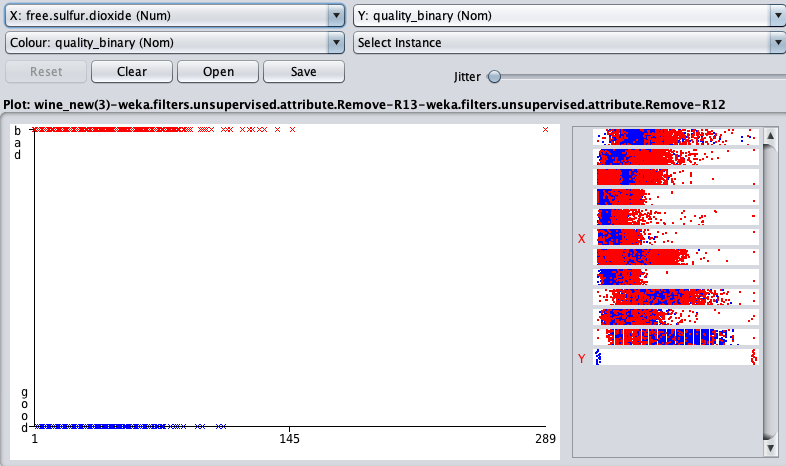
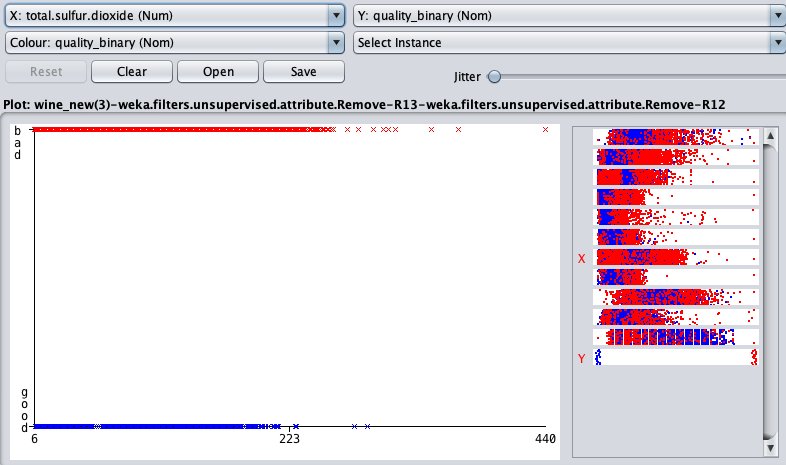
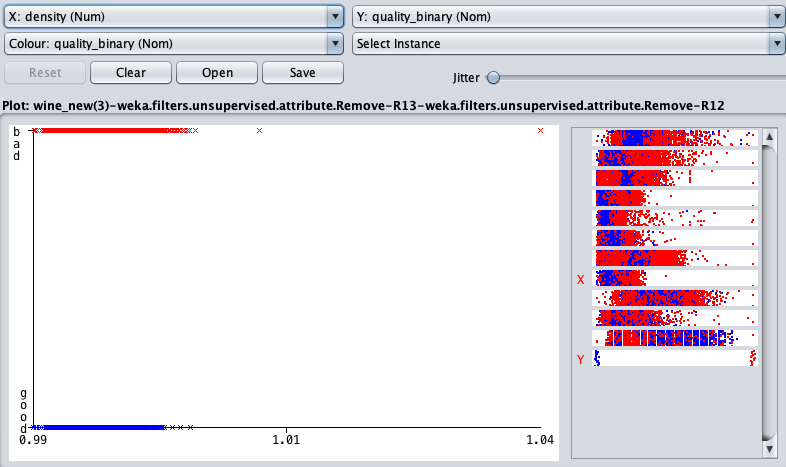
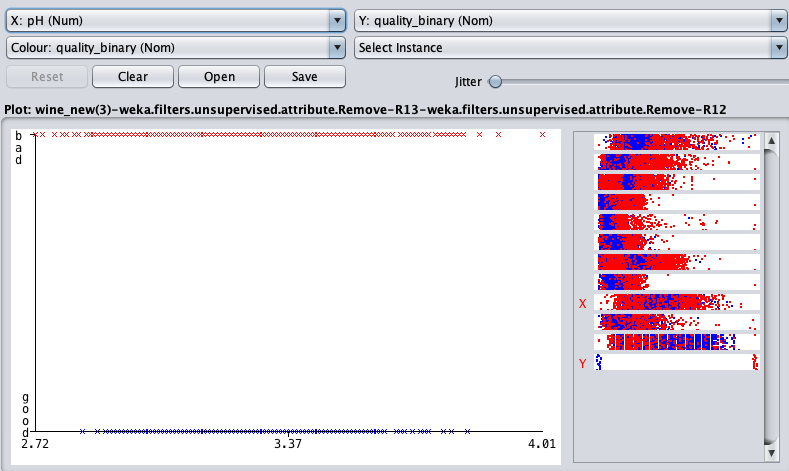
Below is the logistic regression for the wine dataset. I had used the quality of wine which ranged from 3-9 to determine the good vs bad to compare it to other factors like fixed acidity, volatile acidity, pH, density, and etc. As you can see the accuracy of this is 81%.

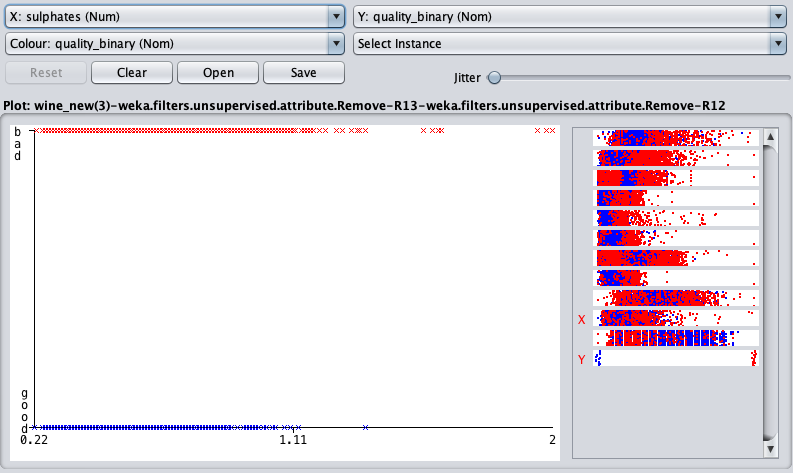
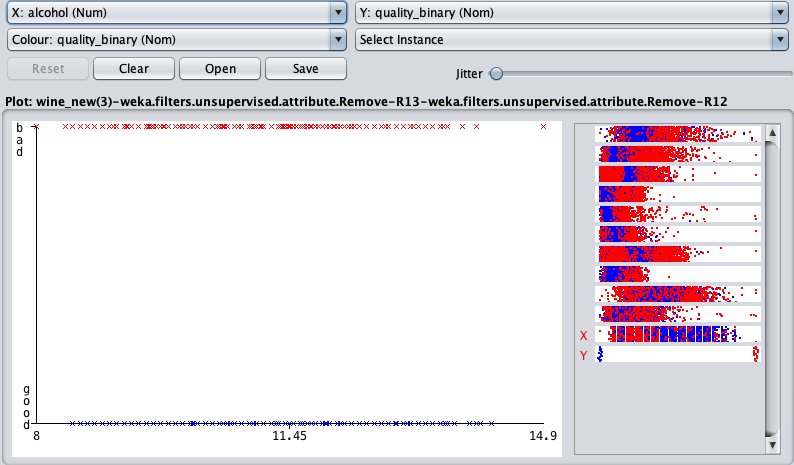


Visualizations:



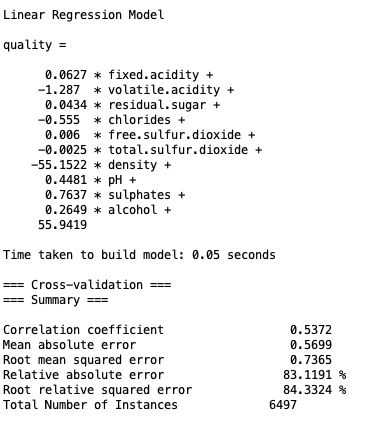
 





Multiple Regression:

The tool, named Linear Regression in WEKA, was used for this dataset could not make a single graph to show all the independent variables like alcohol, pH, etc. vs the dependent variable, which was quality. Below are the graphs for each independent variable vs quality.



# Visualizations:

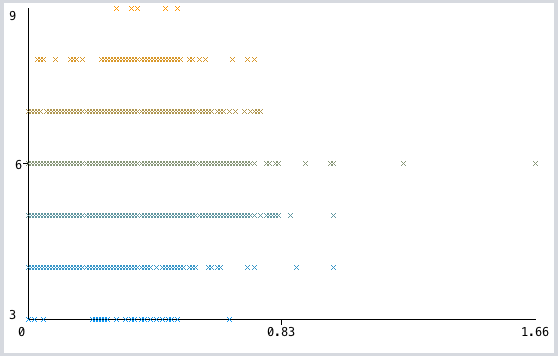
# Quality vs Fixed Acidity

# 

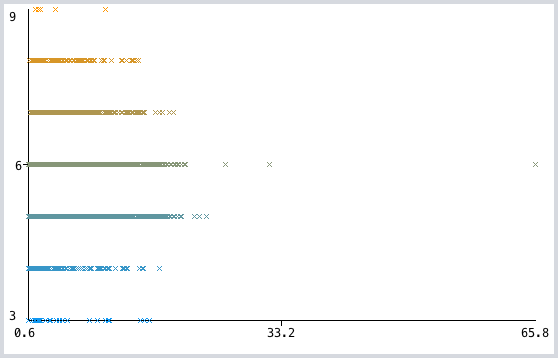
# Quality vs Volatile.acidity

# 

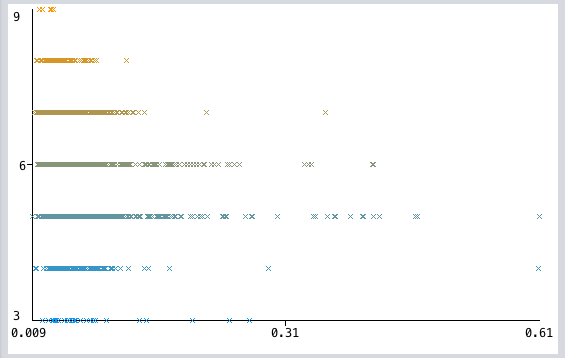
Quality vs Citric.acid



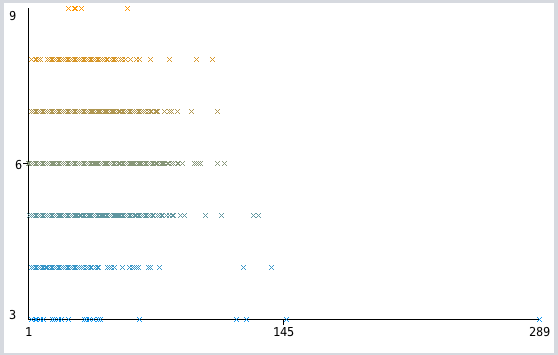
Quality vs Residual.sugar



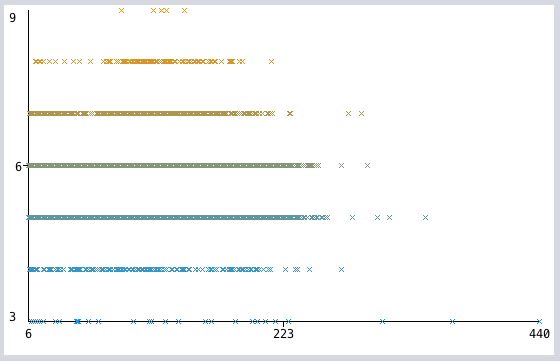
Quality vs. Chlorides



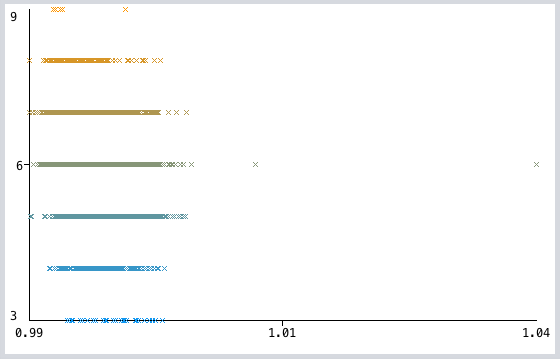
Quality vs Free.sulfur.dioxide



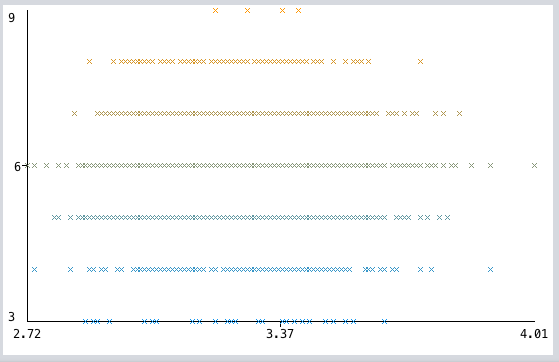
Quality vs Total.sulfur.dioxide



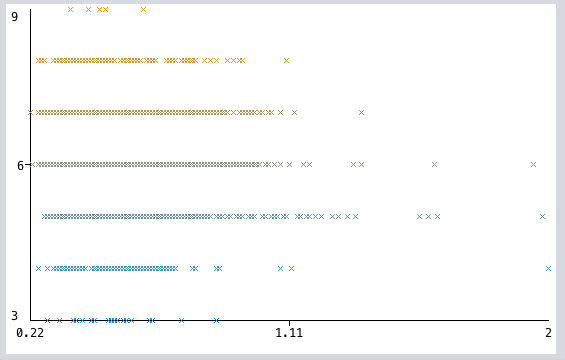
Quality vs Density



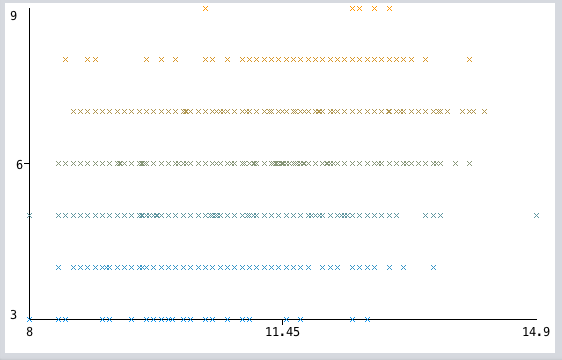
Quality vs pH



Quality vs Sulphates



Quality vs Alcohol



Quality vs Quality

