Ryan Christopher CS677

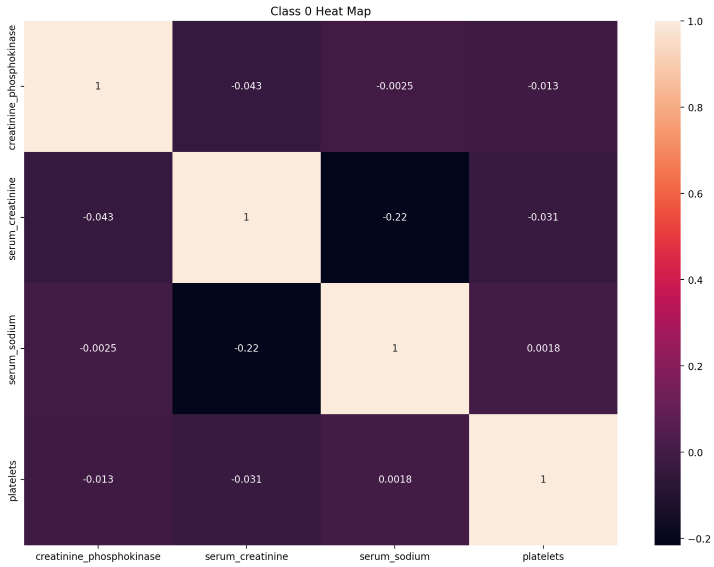
Assignment 4

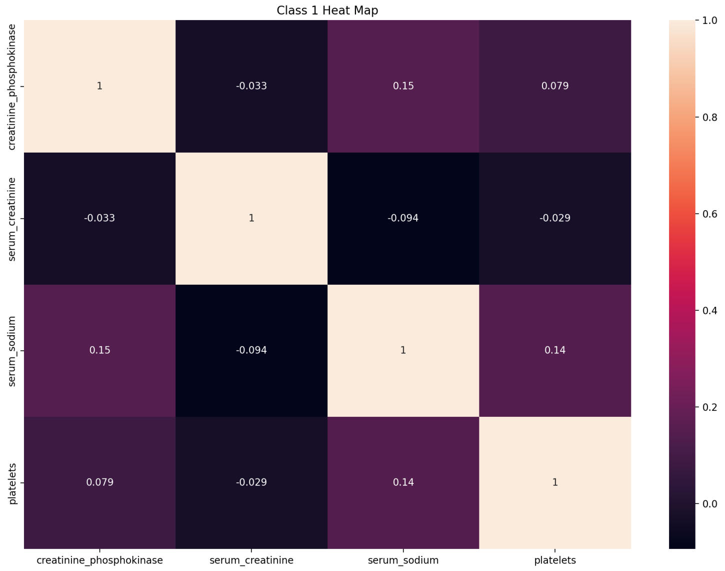
-A New Heart Failure-

Question 1:

Part 1 - createPlots.py loads the data into a dataframe, removes the features that are not being compared, then extracts two dataframes for surviving and deceased patients.

Part 2 - Within createPlots.py, the visual representations of corresponding correlation matrices M0 and M1 are constructed using the pairplot and heatmap methods from the seaborn library. The plots are stored as class_0_heatmap.pdf and class_1_heatmap.pdf within the images folder.





Part 3 -

a) Serum Creatinine and Serum Sodium appear to have the highest correlation in surviving patients.

b) Serum Sodium and Creatinine Phosphokinase appear to have the lowest correlation in surviving patients.

c) Serum Sodium and Creatinine Phosphokinase appear to have the highest correlation in deceased patients.

d) Serum Sodium and Platelets appear to have the lowest correlation in deceased patients.

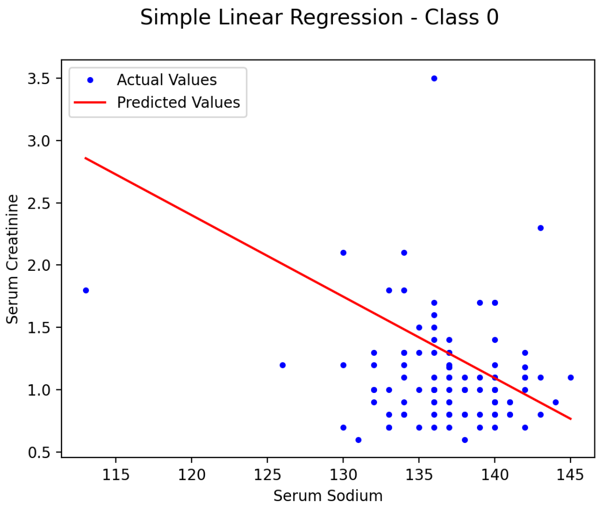
e) No, the results are not the same for both cases. A quality worth noting, however, is the lowest correlation of features in surviving patients is the same as the highest correlation in deceased patients. Both correlations involve the Serum Sodium and Creatinine Phosphokinase features of the patients.

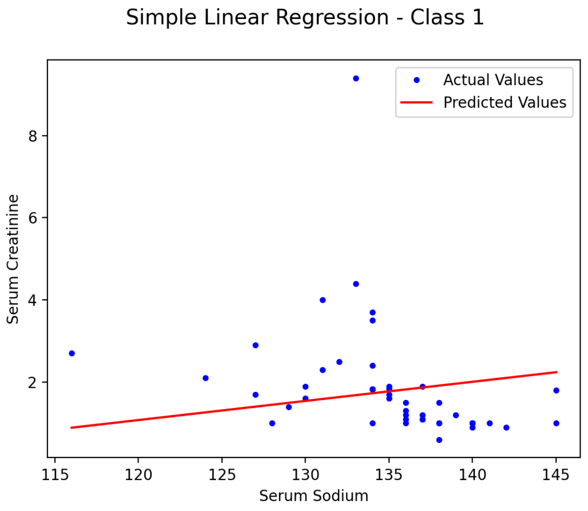
Question 2:

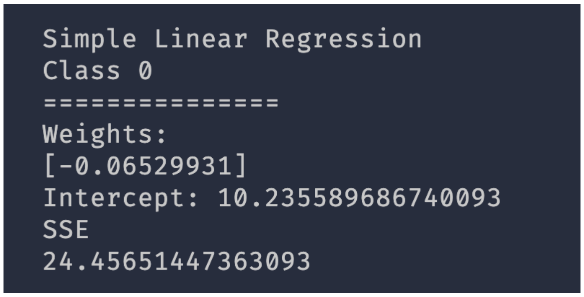
The file applyModels.py takes the two sets of data (surviving and deceased), and using facilitator group 3 (X = serum sodium and Y = serum creatinine) determines the best model from the given list using the applyModel function:

1. $y = ax + b$ - simple linear regression
2. $y = ax^2 + bx + c$ - quadratic
3. $y = ax^3 + bx^2 + cx + d$ - cubic spline
4. $y = alogx + b$ - GLM (generalized linear model)
5. $logy = alogx + b$ GLM (generalized linear model)

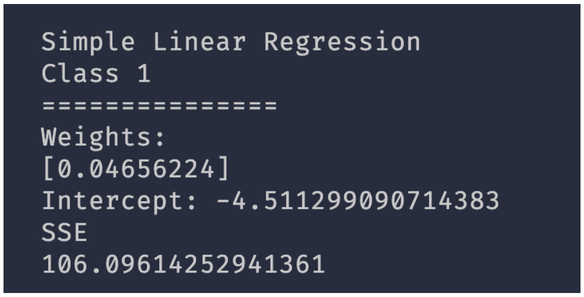
The function runs for both classes on the five models. Each time the function runs, it fits the model on the train data, prints the weights and intercept, computes the predicted values, plots then displays the predicted and actual values, and computes the corresponding loss function (SSE). This can be seen when lines 144 to 146 are run in applyModels.py.


Simple Linear Regression - Class 0

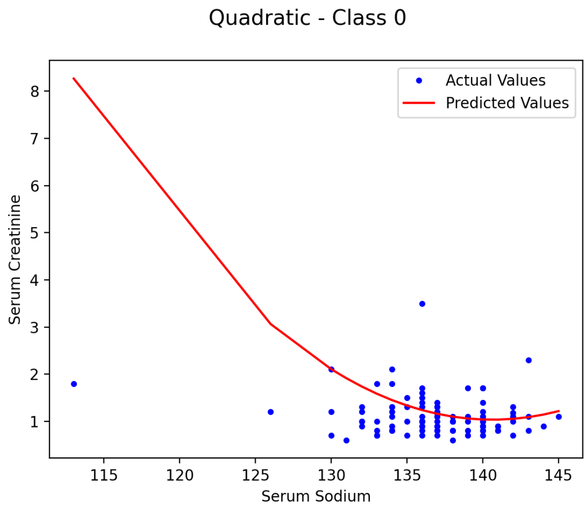

Simple Linear Regression - Class 1

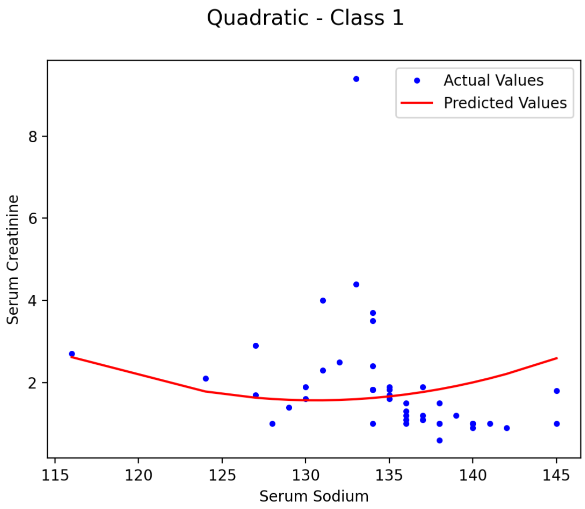


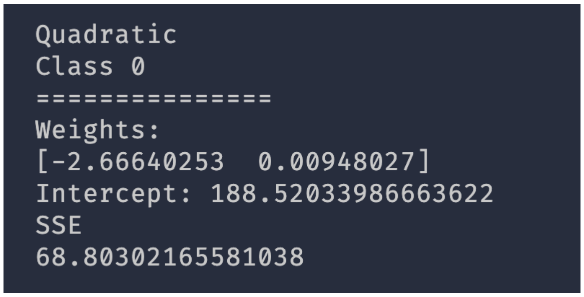
```
Simple Linear Regression
Class 0
===============
Weights:
[-0.06529931]
Intercept: 10.235589686740093
SSE
24.45651447363093
```



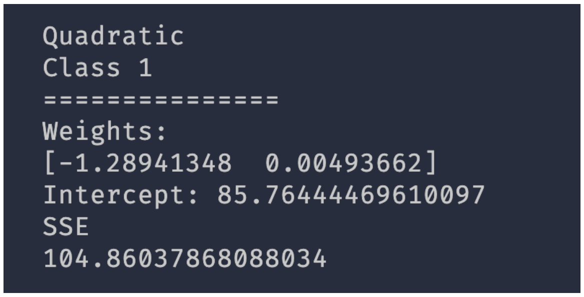
```
Simple Linear Regression
Class 1
===============
Weights:
[0.04656224]
Intercept: -4.511299090714383
SSE
106.09614252941361
```



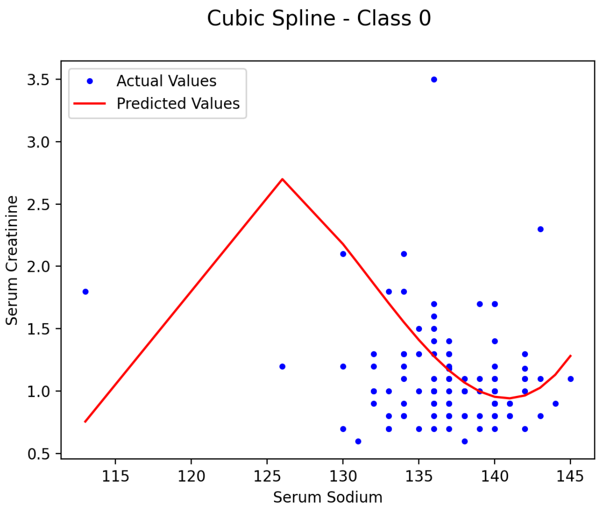


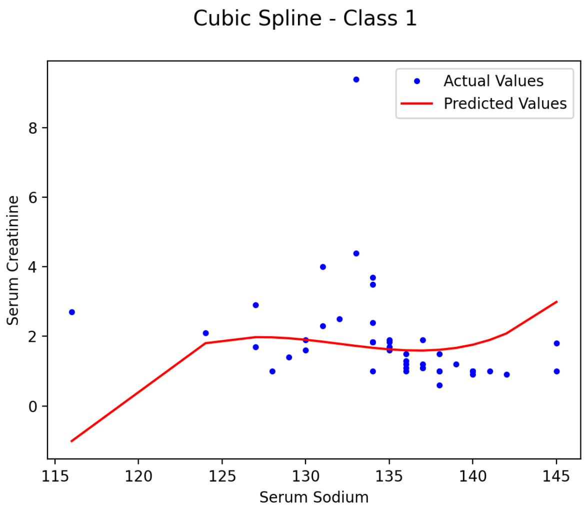


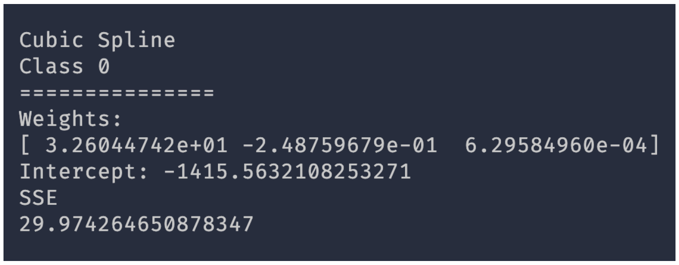
```
Quadratic
Class 0
===============
Weights:
[-2.66640253  0.00948027]
Intercept: 188.52033986663622
SSE
68.80302165581038
```



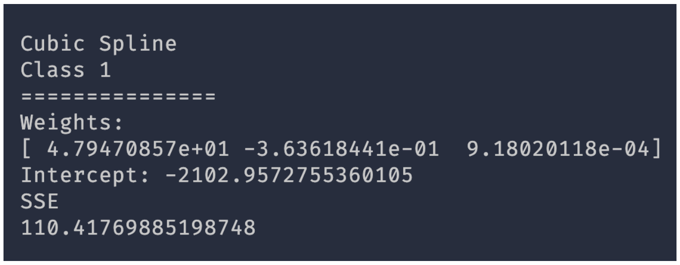
```
Quadratic
Class 1
===============
Weights:
[-1.28941348  0.00493662]
Intercept: 85.76444469610097
SSE
104.86037868088034
```



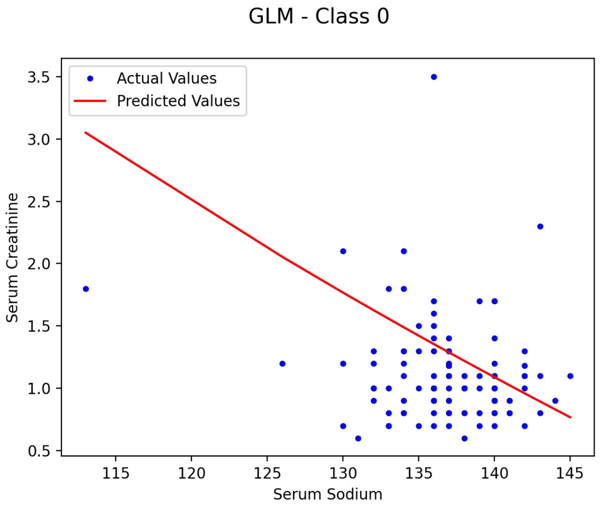


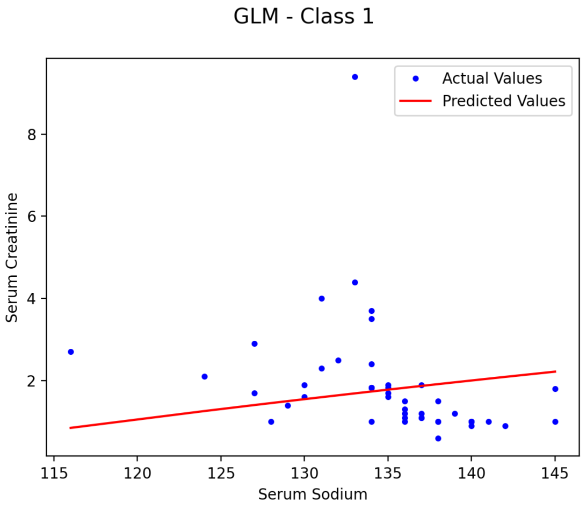


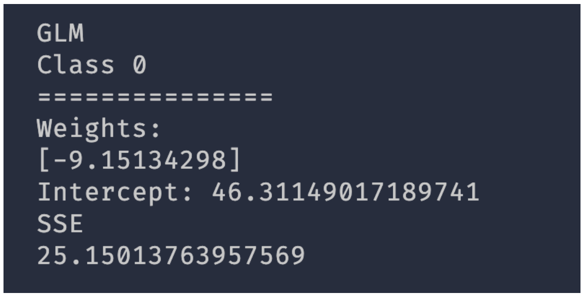
```
Cubic Spline
Class 0
===============
Weights:
[ 3.26044742e+01 -2.48759679e-01  6.29584960e-04]
Intercept: -1415.5632108253271
SSE
29.974264650878347
```



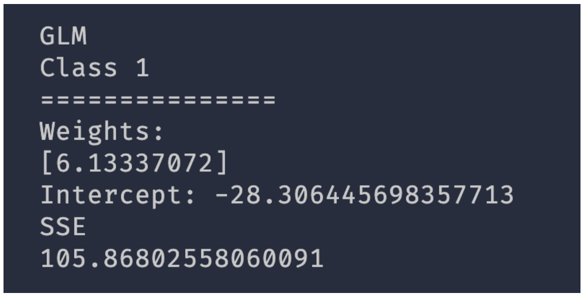
```
Cubic Spline
Class 1
===============
Weights:
[ 4.79470857e+01 -3.63618441e-01  9.18020118e-04]
Intercept: -2102.9572755360105
SSE
110.41769885198748
```



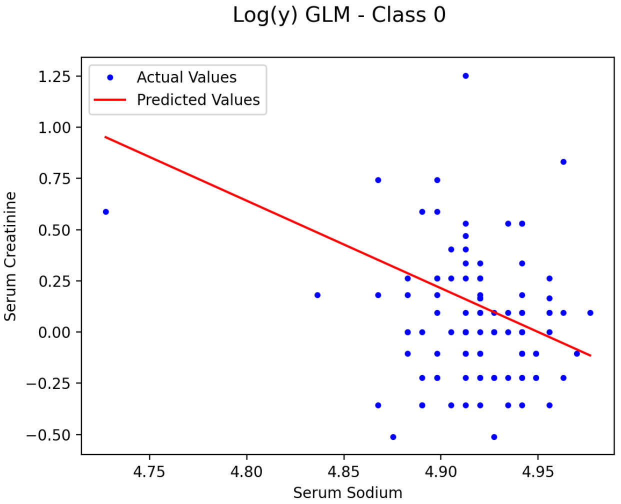


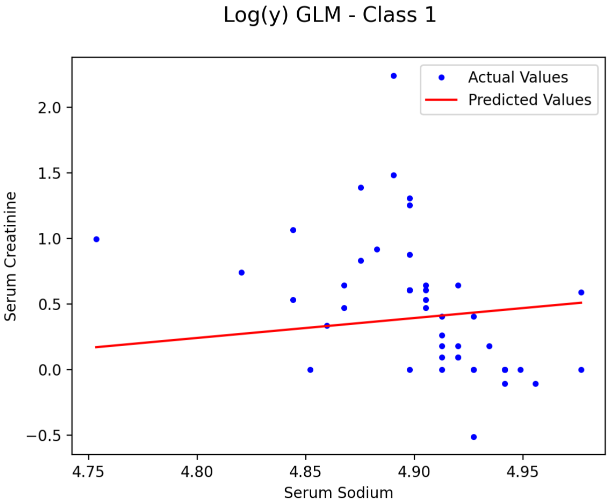


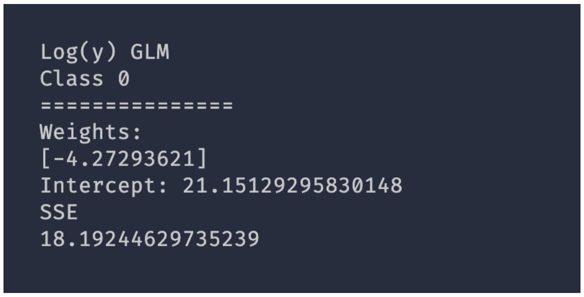
```
GLM
Class 0
===============
Weights:
[-9.15134298]
Intercept: 46.31149017189741
SSE
25.15013763957569
```



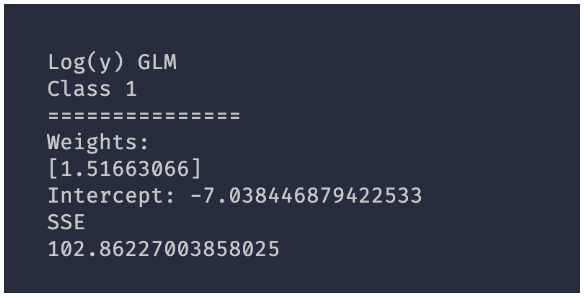
```
GLM
Class 1
===============
Weights:
[6.13337072]
Intercept: -28.306445698357713
SSE
105.86802558060091
```







```
Log(y) GLM
Class 0
==============
Weights:
[-4.27293621]
Intercept: 21.15129295830148
SSE
18.19244629735239
```



```
Log(y) GLM
Class 1
===============
Weights:
[1.51663066]
Intercept: -7.038446879422533
SSE
102.86227003858025
```

Question 3:

*note - all SSE values are rounded to the nearest 2 decimal places. Unrounded values are shown in the outputs when the python files are run.

Model	SSE (death_event = 0)	SSE (death_event = 1)
$y = ax + b$	24.46	106.10
$y = ax^2 + bx + c$	68.80	104.86
$y = ax^3 + bx^2 + cx + d$	29.97	110.42
$y = alogx + b$	25.15	105.87
$logy = alogx + b$	18.19	102.86

Part 1 - The log(y) generalized linear model (model number 5) resulted in the smallest SSE for surviving patients and deceased patients.

Part 2 - The quadratic model resulted in the highest SSE for surviving patients and the cubic spline model resulted in the highest SSE for deceased patients.