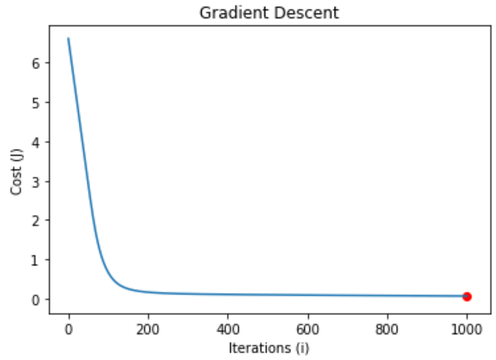
Model Results

- Description of model and testing procedure
    - Description
        - **My model was a simple logistic regression model with gradient descent. To implement the model, I first made a cost function using a matrix implementation. Next, I created a function to calculate weights using gradient descent. Rather than allowing my descent function to converge indefinitely, I opted to specify an iteration parameter for the function. I.e. the caller provides i=1000 to the gradient descent function and weights and costs are calculated 1000 times and the lowest cost/best weights combination is returned. This yielded very successful results, so I stuck with a basic model and did not try to implement quadratic or logarithmic parameters. I also decided against feature scaling sense all of the features were already in a 1 to 4 range (for qualitative data).**
    - Size of your training and test sets.
        - **Training Set (70% of Divorce All) = 118 rows, Test Set (30% of Divorce All) = 52 rows**
    - Initial values that you chose for your weights, alpha, and the initial value for ***J***.
        - **Initial Weights = [-0.43706978 0.10704752 0.2838941 0.08700884 0.05469661 0.39459196 -0.11364312 0.9790714 0.1422245 0.11009641 -0.25255551 0.42538224 -0.01907929 0.03766554 0.10085491 -0.07539902 0.62458694 0.23807971 0.59428697 0.07097317 0.41124834 0.41412357 0.24591391 0.53951566 -0.13121335 -0.20979962 0.10965417 0.01250845 0.81625906 0.17850526 0.14343865 0.05077078 -0.49056577 0.41399329 0.2573406 0.04117911 0.94208505 -0.0225656 0.33215277 0.47397586 0.33750888 0.01325591 0.18548625 -0.28060235 0.37678714 -0.1344187 -0.50720569 -0.3944416 -0.29920594 -0.0703983 -0.7240705 -0.36627092 -0.50006548 -0.05297404 -0.02733274]**
        - **Initial Alpha = 0.1**
        - **Initial Iterations = 100**
        - **Initial training value for J = 0.05047597424776233**
    - Final values for alpha, your weights, how many iterations your learning algorithm went through and your final value of ***J*** on your training set.
        - **Final Weights = [1.18852099e-02 2.61429460e-02 1.73358624e-01 -2.68390942e-01 1.45514569e-04 1.39831754e-01 -3.52120595e-03 4.74251355e-02 3.16425113e-01 6.78225513e-01 -2.58552298e-01 7.06436713e-01 3.62691362e-01 2.50002940e-01 5.90311102e-02 3.62969774e-01 7.82335182e-01 2.12115567e-01 5.58059382e-01 2.83042527e-01 8.86379428e-01 -1.71695219e-03 3.47074246e-02 8.43034745e-01 -1.26726617e-01 7.87633128e-02 9.25503504e-01 1.28984154e-01 4.13987140e-01 5.88769820e-01 8.97878608e-01 -3.76130840e-01 8.50125198e-02 8.59316142e-02 1.51148061e-01 6.69491023e-01 7.02223514e-01 1.31639815e-01 -9.40497210e-02 5.84754137e-01 3.78257489e-01 2.70665274e-01 -2.53019250e-01 -7.03238928e-01 -3.43960954e-02 5.93490966e-03 -6.60041556e-01 -3.86597714e-01 -8.01347651e-01 -5.12348686e-01 1.43809465e-01 -8.02018417e-01 -1.06433430e-01 -2.37929589e-01 -2.64180905e-02]**
        - **Fina Alpha = 0.01**
        - **Final Iterations = 1000**
        - **Final training value for J = 0.051263823063813885**

    - Include a plot of ***J*** (vertical axis) vs. number of iterations (horizontal axis).



        -
    - If you did feature scaling, describe what you did.
        - **I elected not to do feature scaling as all features were already in the range of 1 to 4**
    - Value of ***J*** on your test set.
        - **Final value for J = 0.04735597524524438**
- A confusion matrix showing your results on your test set.

	Predicted Yes	**Predicted No**
Actual Yes	**TP = 20**	**FN = 1**
Actual No	**FP = 0**	**TN = 31**

- A description of your final results that includes accuracy, precision, recall and F1 values.
    - **Accuracy = 0.9807692307692307**
    - **Precision = 1.0**
    - **Recall = 0.9523809523809523**
    - **F1 Score = 0.975609756097561**