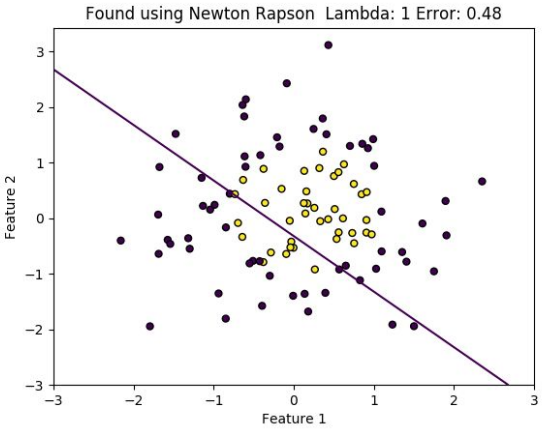
Logistic Regression model analysis

Our Logistic regression model is trained through iterative gradient descent and Newton Raphson method. The program is implemented in logistic.py.

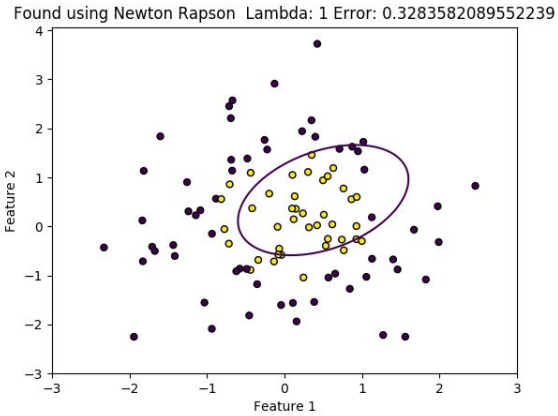
Approach and graphs plotted

1. Plot dataset for representation of data.
2. Train model with gradient descent and Newton Raphson method. Graphs for both methods are generated in folders newton and grad for degree 2, 3 and 4.
3. Plot polynomial decision boundary for overfit and underfit data.

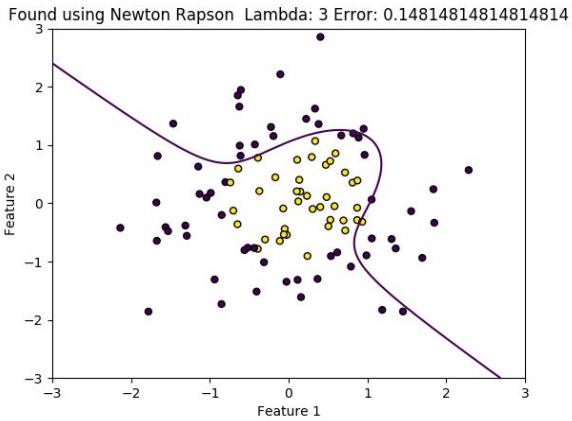
Plots using Newton Raphson



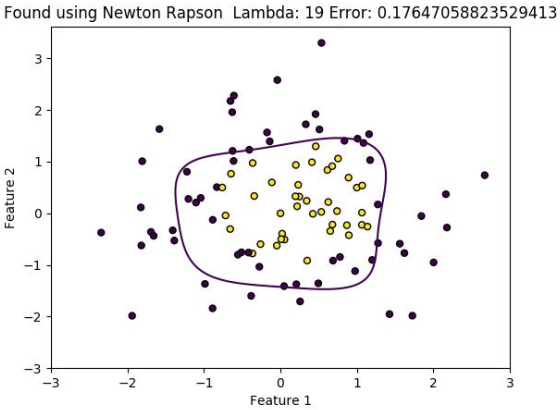
Degree = 1



Degree = 2

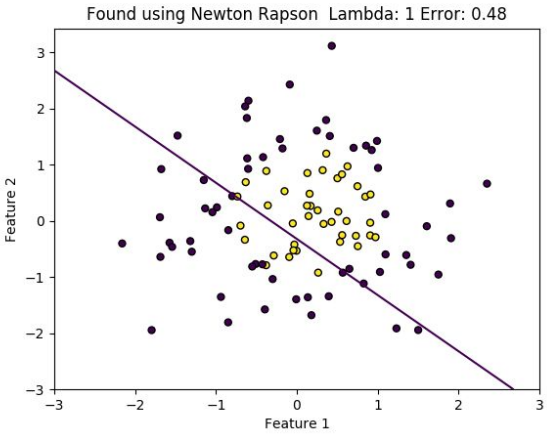


Degree = 3

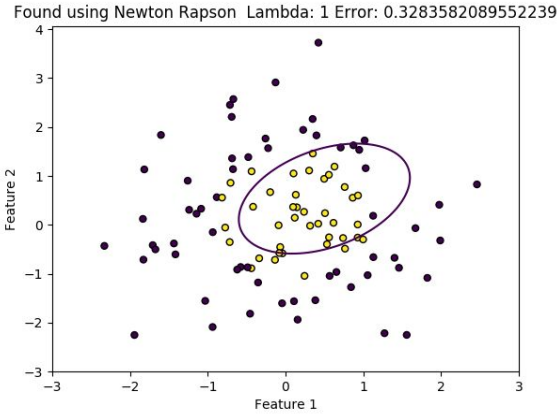


Degree = 4

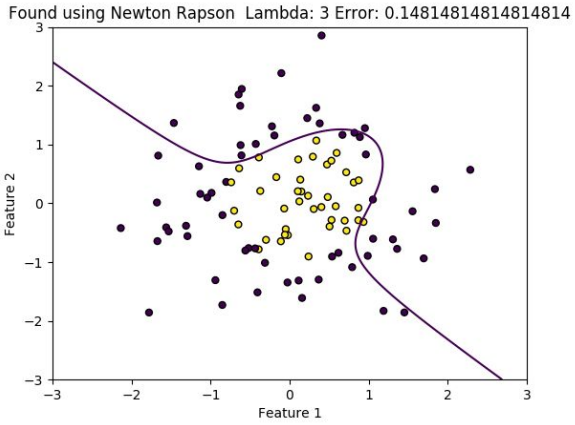
Plots using Gradient Descent



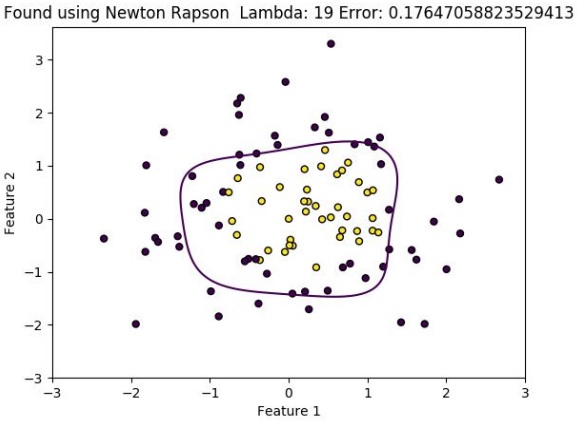
Degree = 1



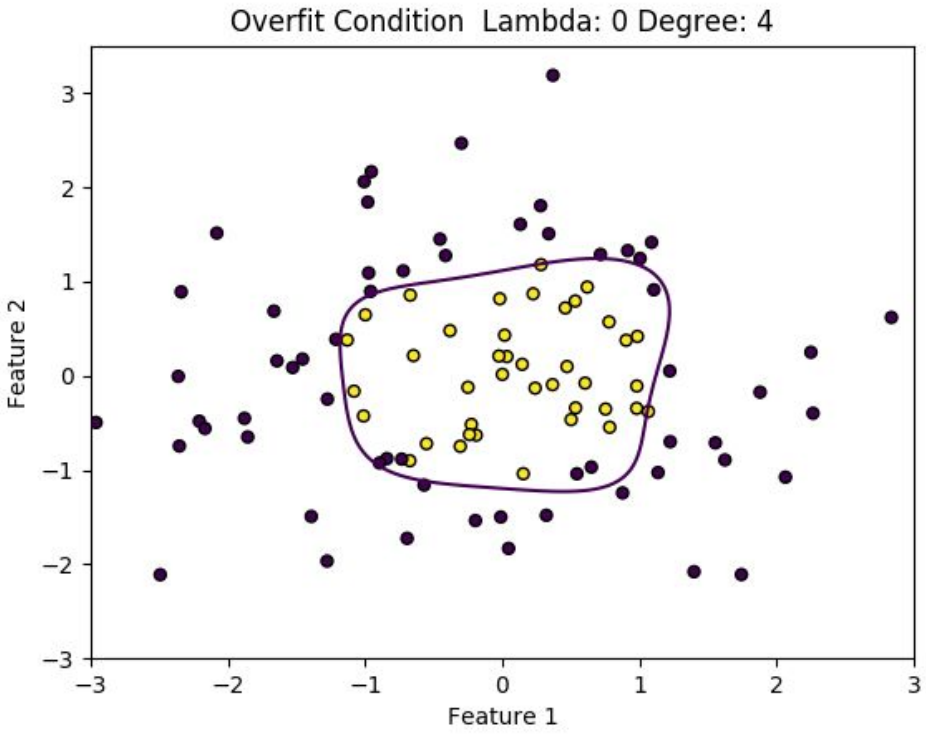
Degree = 2

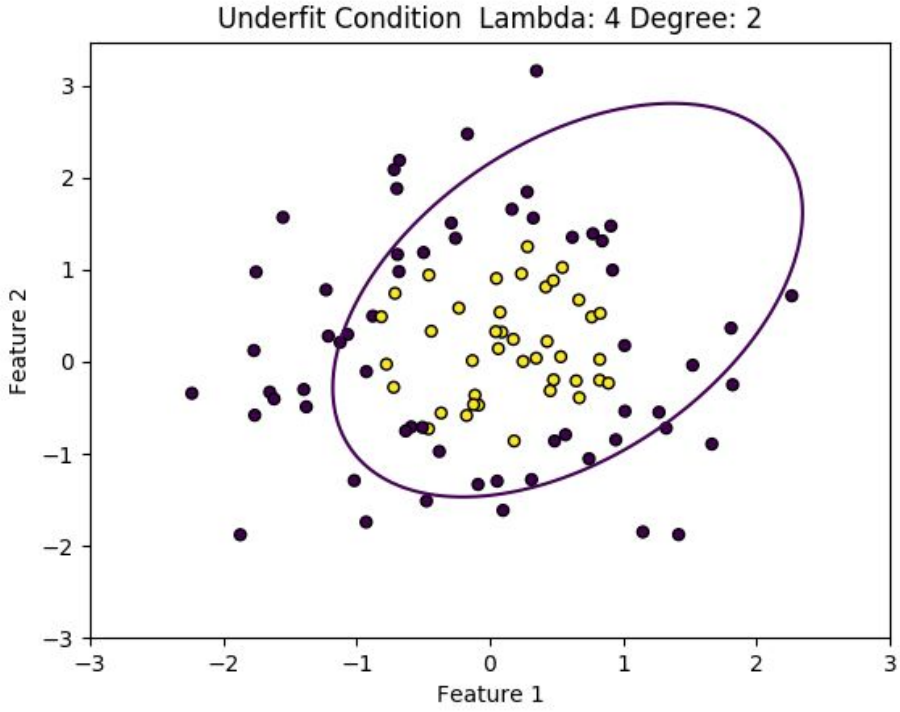


Degree = 3



Degree = 4


Overfit Condition Lambda: 0 Degree: 4
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3


Underfit Condition Lambda: 4 Degree: 2
Feature 2
Feature 1
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Observations

1. Newton Raphson takes very less iterations about 20 to converge.
2. Gradient descent takes many iterations, I used 10000 iterations.
3. There is a huge overhead in each iteration of Newton Raphson and it is very sensitive to initial values of theta.
4. Therefore gradient descent is much more preferred optimization technique.