For lambda equal to zero you get the best accuracy however the circle is not very uniform and tries to avoid the not accepted parameters. For lambda equal to 1 you get a relatively uniform circle that takes mostly the correct values but also has a few bad ones. For lambda equal to 10 it accepts almost everything regardless of whether it is good or bad and only has a 50% accuracy. The value of 100 accepts absolutely everything the boundary line can no longer be seen and the accuracy is a grand total of 7%. Lamda regulates the boundary and the higher the lambda value the less tightly bound the constraints leading to lower accuracy. Lamda of 1 gives the best overall circle however lambda 0 gives the best accuracy and data partitioning however it appears to be slightly overfitted, Therefore I would go with 1.

Problem 3:
The different parameters found in the sklearn logistic regression function are penalty, dual, tol, c, fit_intercept, intercept, scaling, class_weight, random_state, solve,r max_iter, multi_class warm_start, verbose, n_jobs, l1_ratio,

Penalty: affects the penalty result and the type of penalty term.
Dual: type of formulation for l2 penalty and liblinear solver.
Tol: tolerance for the stopping criteria
C: affects regularization the smaller the value the stronger the regularization
fit_intercept: sets if a constant needs to be added in the decision-making function.
Intercept_scaling: used in liblinear to append instant vector.
Class_weight: assigns a weight value to determine the importance of a class to the learning process. Adjusts how focuses the different class labels will be.
random_state: how the data is shuffled
Solver: type of algorithm used to optimize the data set. Depending on the size some are better than others
Max_iter: is the number of iterations to converge.
Multi_class: controls loss minimizations and problem type
Verbose: affects output results
N_jobs: number of cores that are used in parallel
l1_ratio : affects the penalty parameters for elasticnet

The parameters for the sklearn logistic function affect the different ways the model processes its contents, data, and solver constants. Depending on the data you use you can mix and match settings to obtain more optimal results. You can also increase and decrease iterations to find more optimal solutions and fix the best fit.