MACHINE LEARNING

ECE 4332 / ECE 5332

Implementation of Logistic Regression Classification

Spring 2019 – Project 4

Team:

Shubham Trehan

Shuvalaxmi Dass

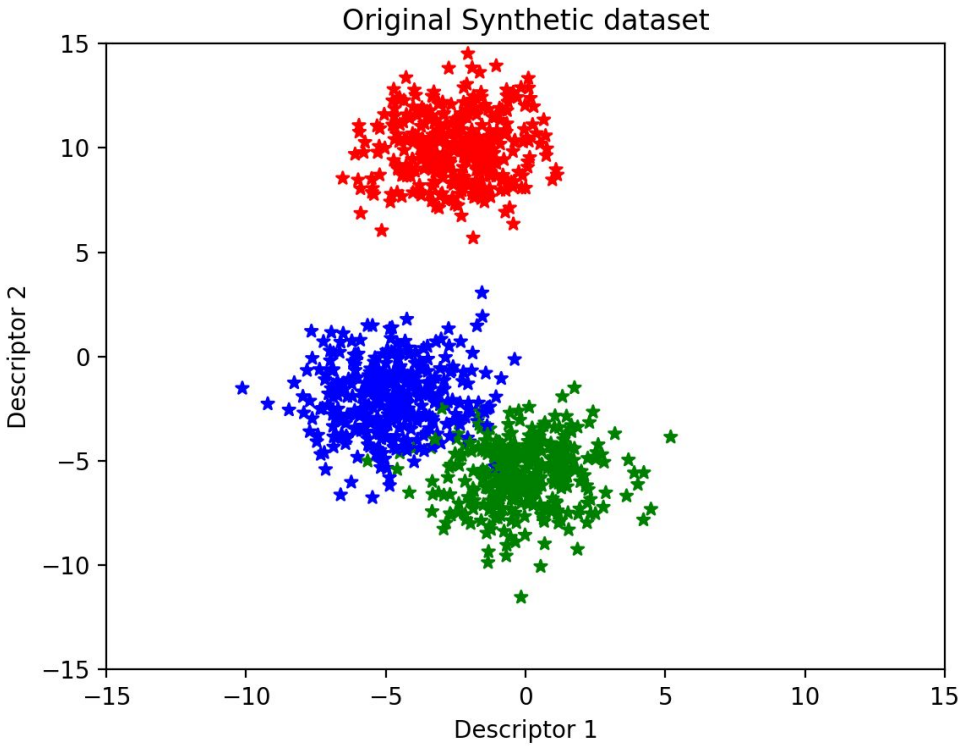
Sraddhanjali Acharya

A. Validation of our implementation using synthetic data:

We created synthetic data using sklearn.make_blobs from ***Scikit-Learn library***, with parameters number of samples, number of features, number of centres (classes), cluster's standard deviation and a random_state.

Parameters changed to construct our synthetic data:

1. Number of samples per class: 1000
2. Number of classes: 3 (Class0, Class1 and Class2)



Graph of original synthetic data

Model Information:

i) Basis function: Polynomial (m = 1, 2, 3…)

ii) Learning rate: (L_rate = 0.001, 0.05, 0.5)

iii) Regularization Parameter: (lambda = 0.001, 0.1, 0.005)

Best Model for Train-Test Split (Hold-Out method):

Train split (80%), Test split (20%)

i) m = 3

ii) L_rate = 0.05

iii) Lambda = 0.00001

Accuracy and Times

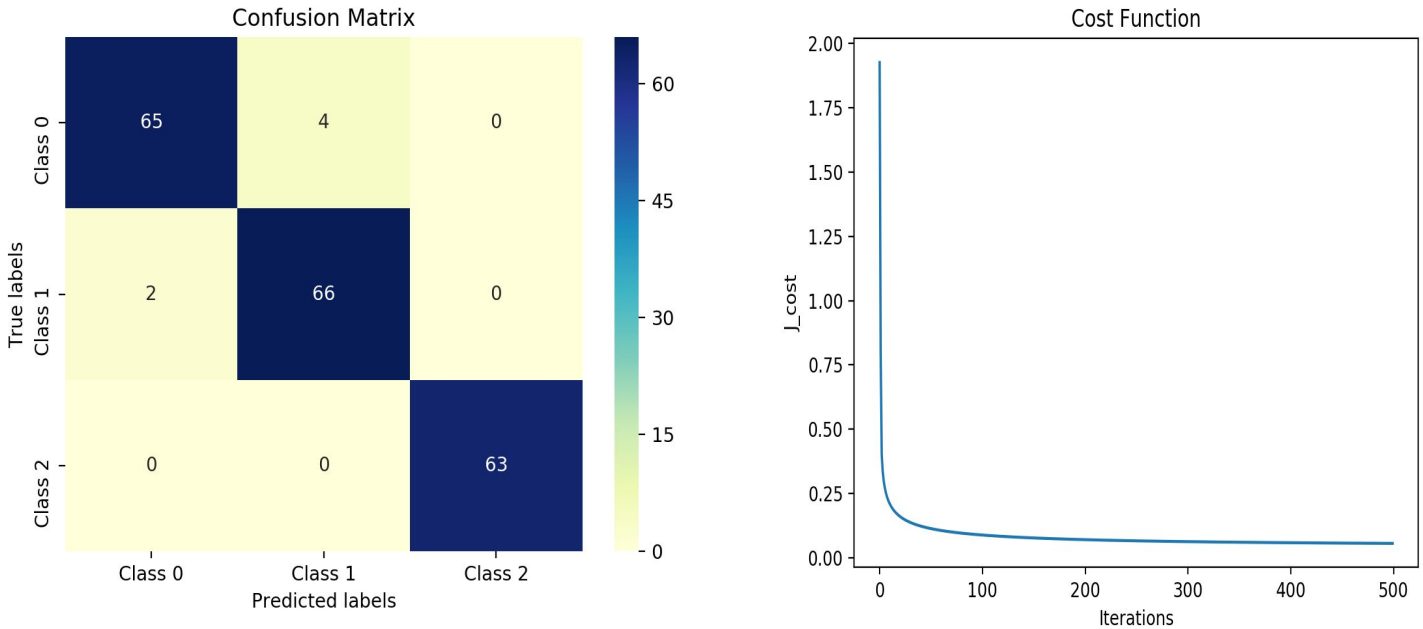
i)Train Accuracy = 98.75%

ii) Test Accuracy = 97%

iii)Training time = 15.7 seconds

iv)Testing time = 8.98e-05 seconds

We can see from the confusion matrix the blue colour shows the number correctly classified observation by our Logistic Regression model.Moreover, The J_cost graph shows how that it is decreasing as we keep increasing the iteration number.



Confusion matrix and Cost Function Convergence for our best model

K-Fold Cross-Validation

We checked the mean accuracy for train and test for K-folds= [5, 8, 10, 15, 20] keeping the rest of the hyperparameters same as that of the best model mentioned before. We have reported the results for K = 20 as this model not only had the least training and testing times amongst all K values aforementioned, the training error and testing error are close to one another. This means the model we trained is either not overfitted or under-fitted, but a good enough model.

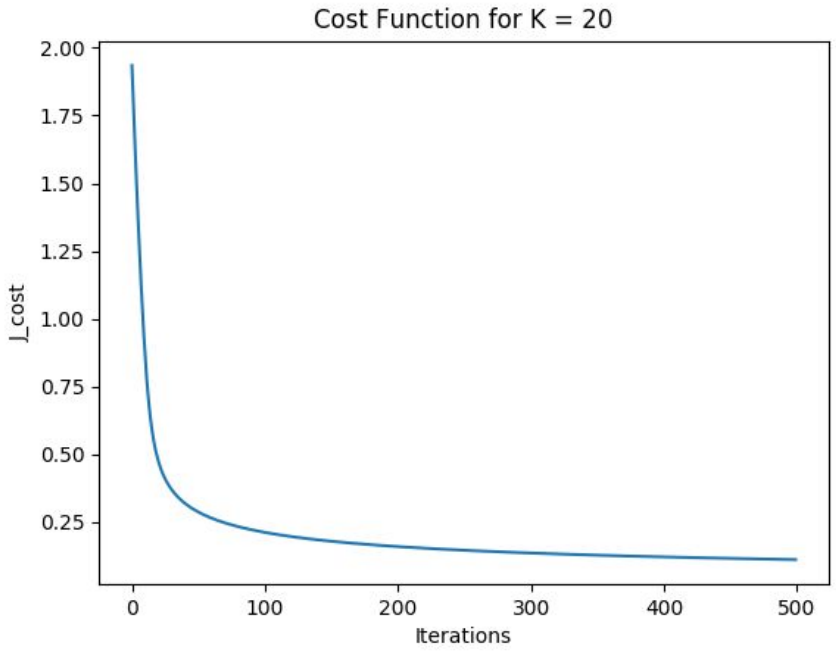
K-Fold Cross-Validation for K = 20

i)Mean Accuracy (Train) = 97.321%

ii)Mean Accuracy (Test) = 97.3%

iii)Mean Training Time = 14.3 seconds

iv)Mean Testing Time = 1.95 e-05 seconds



Cost Function Convergence for our best model

B. Description of selected, real-world dataset:

We picked **Iris Dataset** with 3 iris flower species (multiclass problem) to classify them based on their features.

Iris Dataset Information:

i) Number of classes: 3

ii) Number of samples per class: 50

iii) Class Names: Iris Setosa, Iris versicolour, Iris Virginica

iv) Features: Sepal length, Sepal width, Petal length, and Petal width in cms.

C. Systematic evaluation of the performance of logistic regression on selected dataset and comparison to reported benchmark(s):

Model Information:

i) Basis function: Polynomial (m = 1, 2, 3…)

ii) Learning rate: (L_rate = 0.001, 0.05, 0.5)

iii) Regularization Parameter: (lambda = 0.001, 0.1, 0.005)

Best Model for Hold-Out method:

Train split (80%), Test split (20%)

i) m = 4

ii) L_rate = 0.5

iii) Lambda = 0.00001

Accuracy and Times

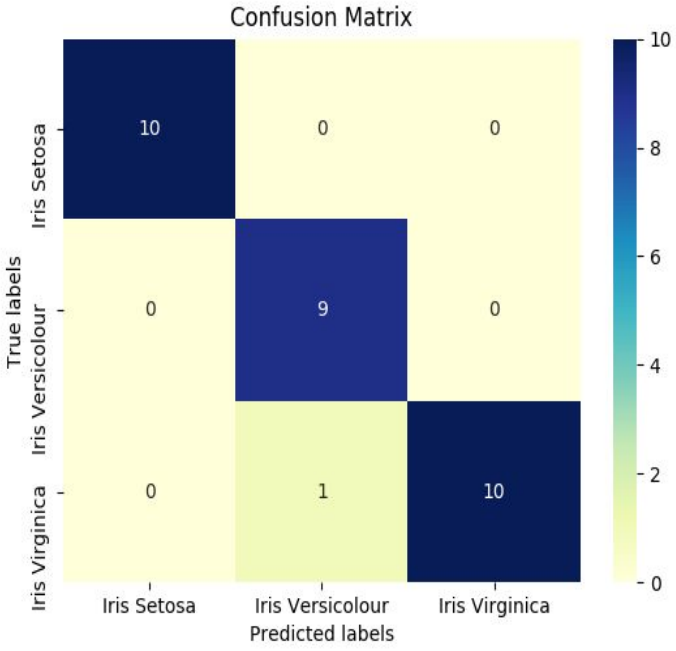
i) Train Accuracy = 98.75%

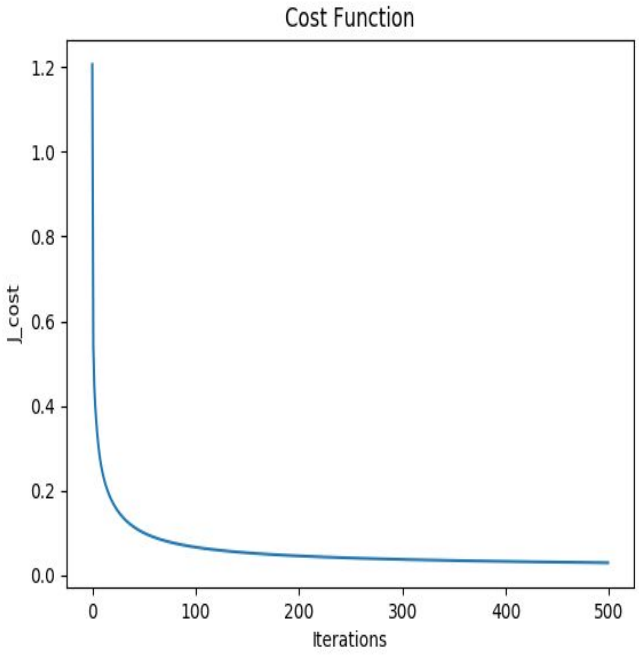
ii) Test Accuracy = 97%

iii) Training time = 2.23 seconds

iv) Testing time = 1.1 e-4 seconds

Confusion matrix and Cost Function Convergence for our best model





Similarly to the synthetic dataset, we only report results for K=20 as it has the least training and testing times as compared to other K values (5, 8, 10, 15) we tried.

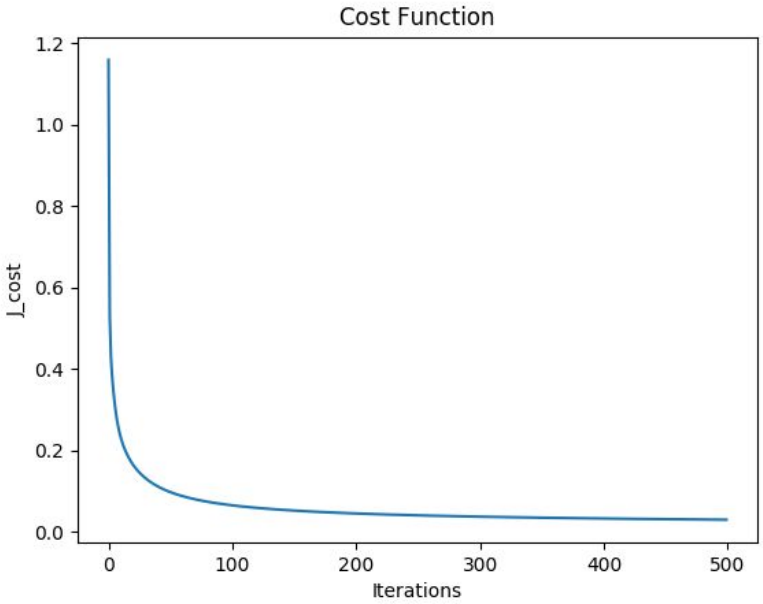
K-Fold Cross-Validation for K = 20.

i) Mean Accuracy (Train) = 98.702%

ii) Mean Accuracy (Test) = 96.16%

iii) Mean Training Time = 2.305 seconds

iv) Mean Testing Time = 2.28 e-05 seconds



Cost Function for the K=20

Benchmark:

We compare our results with [1] as a benchmark. Results section in [1] show accuracies of 99.225% for training and 100 % for testing with a Radial basis function network (neural network). In comparing with our K=20 cross-validation method where mean accuracies for training and testing are 98.702% and 96.16%; our training accuracy is close to the benchmark, however, testing accuracy is behind by about 3.84%.

D. Training and testing times:

We observe a larger training time for synthetic data (almost 7 times) compared to Iris for getting the best model. This can be explained by the larger number of samples per class in our synthetic data compared to our Iris dataset. Despite synthetic data having fewer features than Iris dataset, it seems larger dataset size might have affected the training and testing times.

For the Hold-Out method, synthetic dataset took *15.7* seconds for training and *8.98e-05* seconds for testing. Whereas Iris dataset took *2.23* seconds for training and *1.1e-4* seconds for testing.

For K-fold (K= 20) cross-validation method, synthetic dataset took *14.3* seconds

for training and *1.95 e-05* seconds for testing. Whereas in Iris dataset, the training and testing times were *2.305* and *2.28 e-05* seconds respectively.

Therefore, in both methods, we can conclude that the Synthetic dataset took longer time than Iris dataset. This may be because of the larger samples in synthetic dataset compared to Iris.

REFERENCE:

1. http://lab.fs.uni-lj.si/lasin/wp/IMIT_files/neural/doc/seminar8.pdf