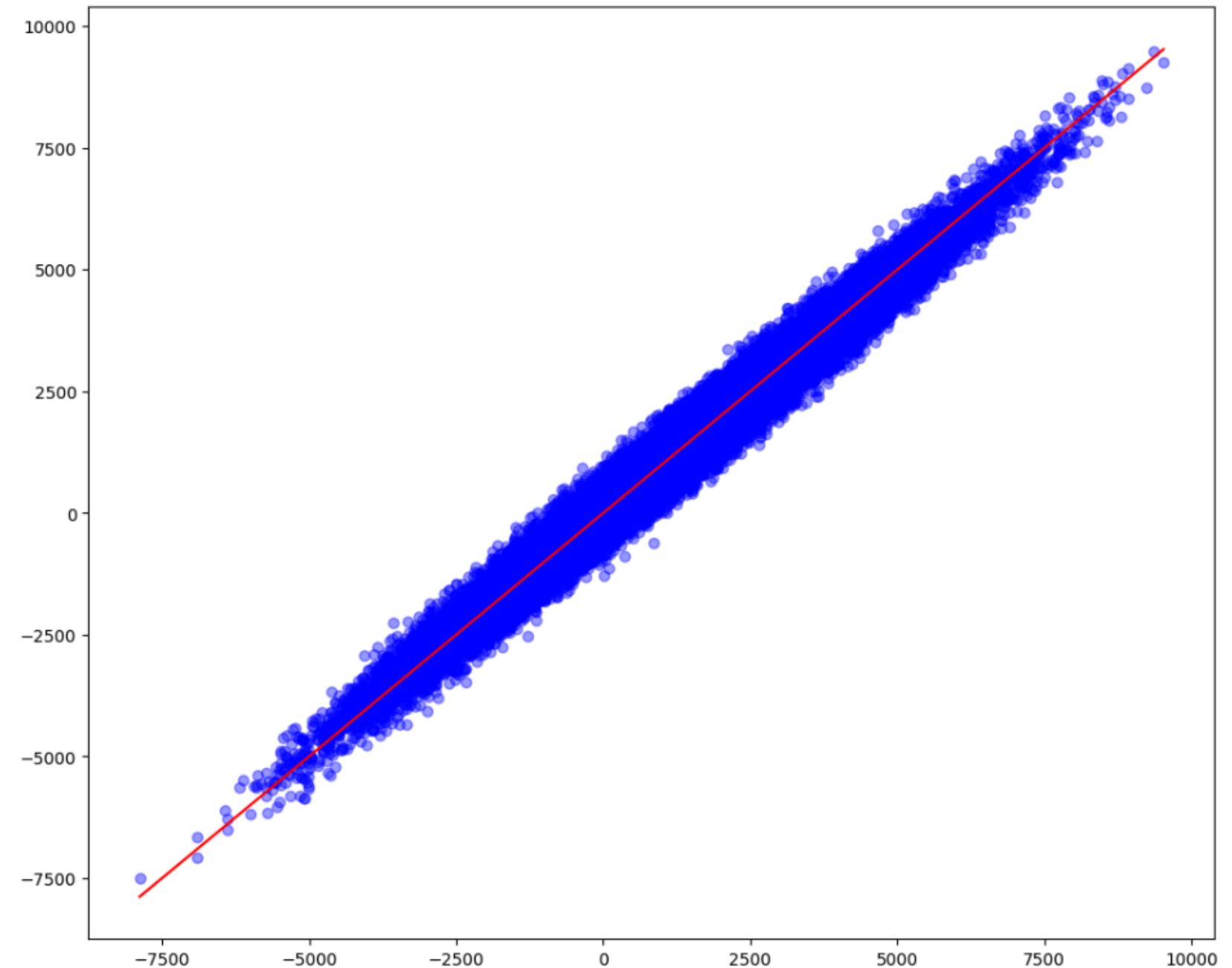


Linear Regression

Gradient Descent and also tried Ordinary Least Squares but was not very effective as dataset was large

Mean Squared Error: 0.0249
Mean Absolute Error: 272.6110
R-squared: 0.9751



Prediction vs Actual Plot

Polynomial Regression

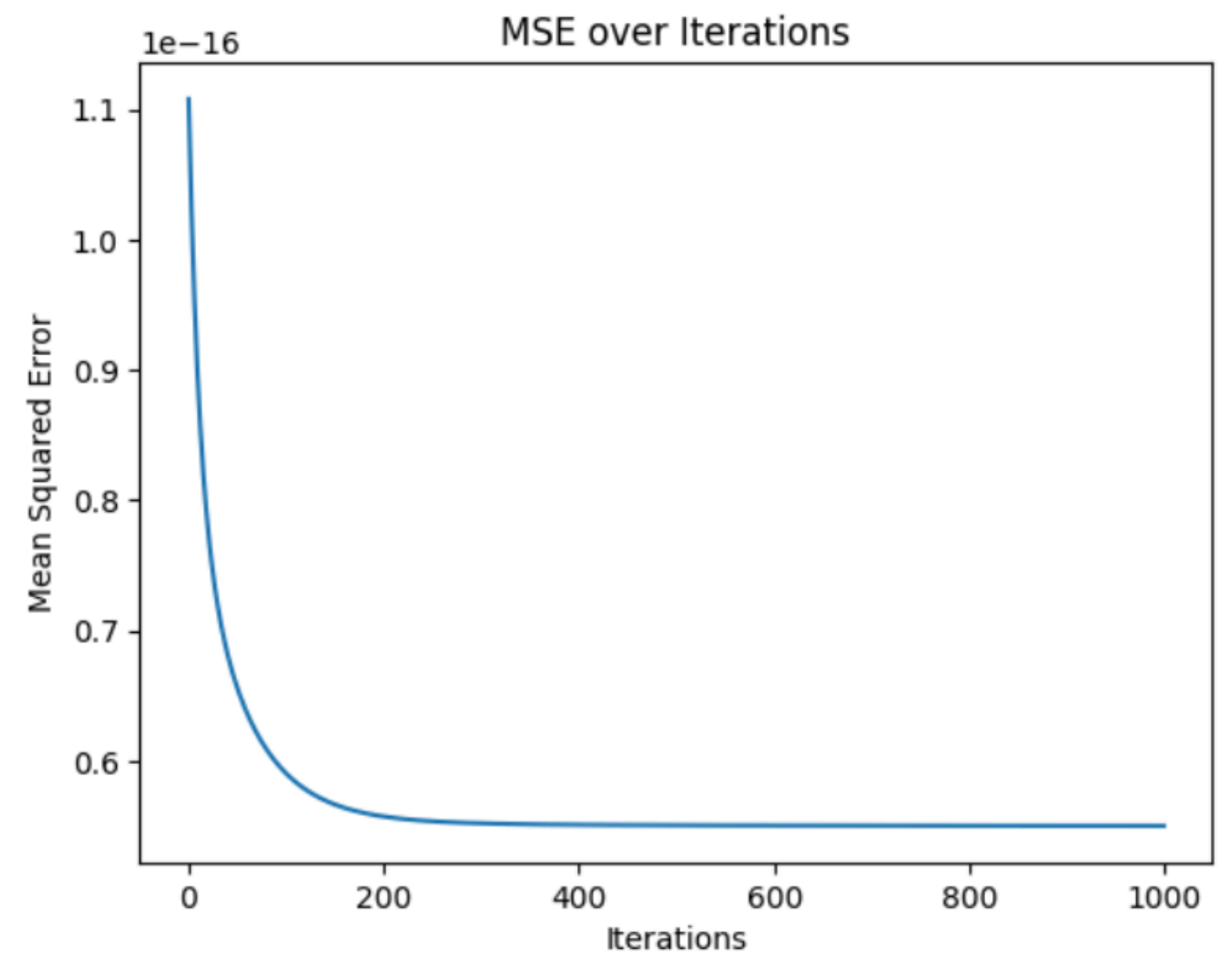
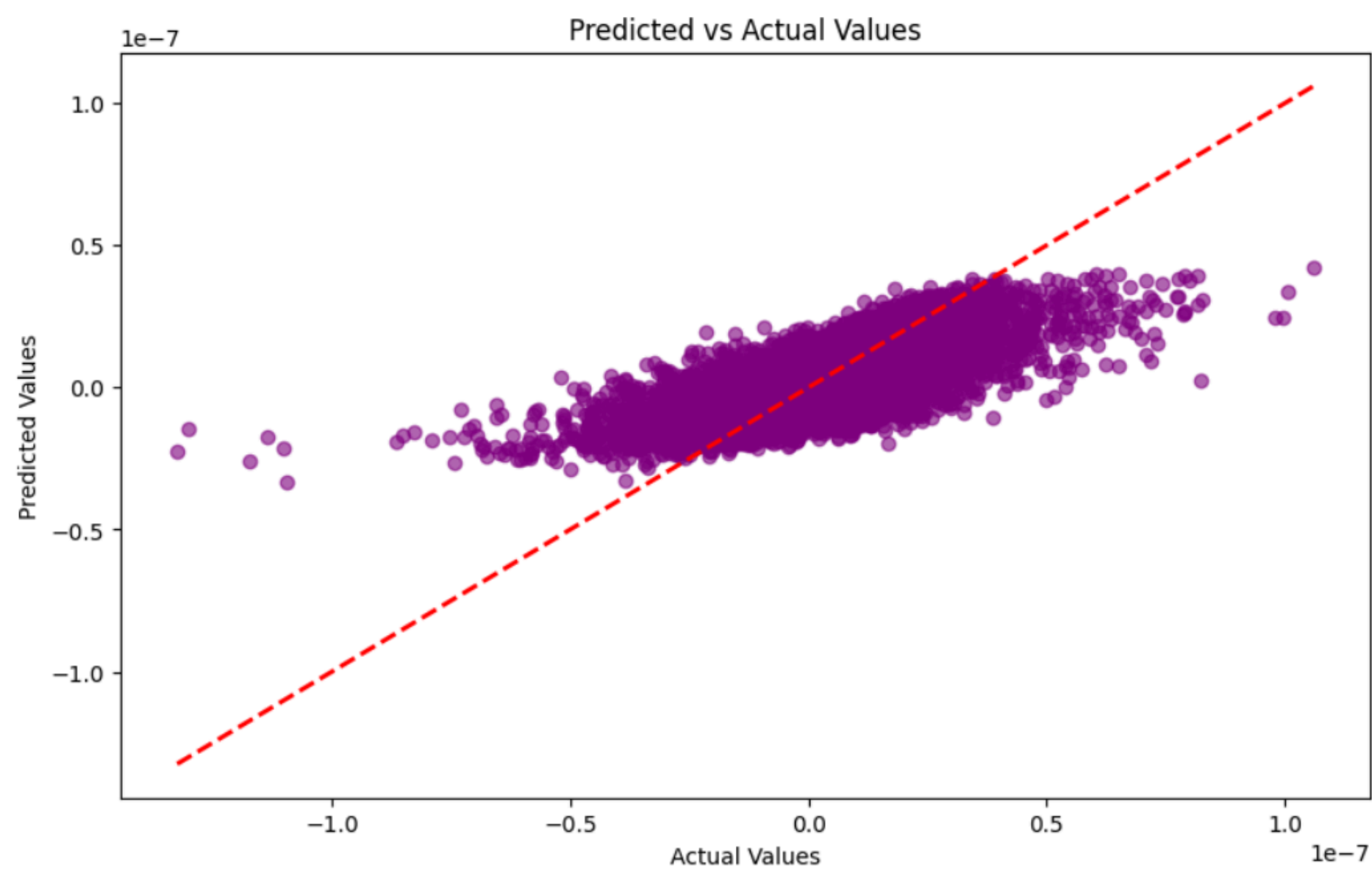
Gradient Descent and also (OLS) but it was very vague for a large dataset,
caused underfitting with low accuracy

Final MSE: $5.502e-17$

Mean Absolute Error: $4.84162e-9$

R-squared: 0.5177387206657568

Unique thing to be noted is that b_0 is always 0



Binary Classification

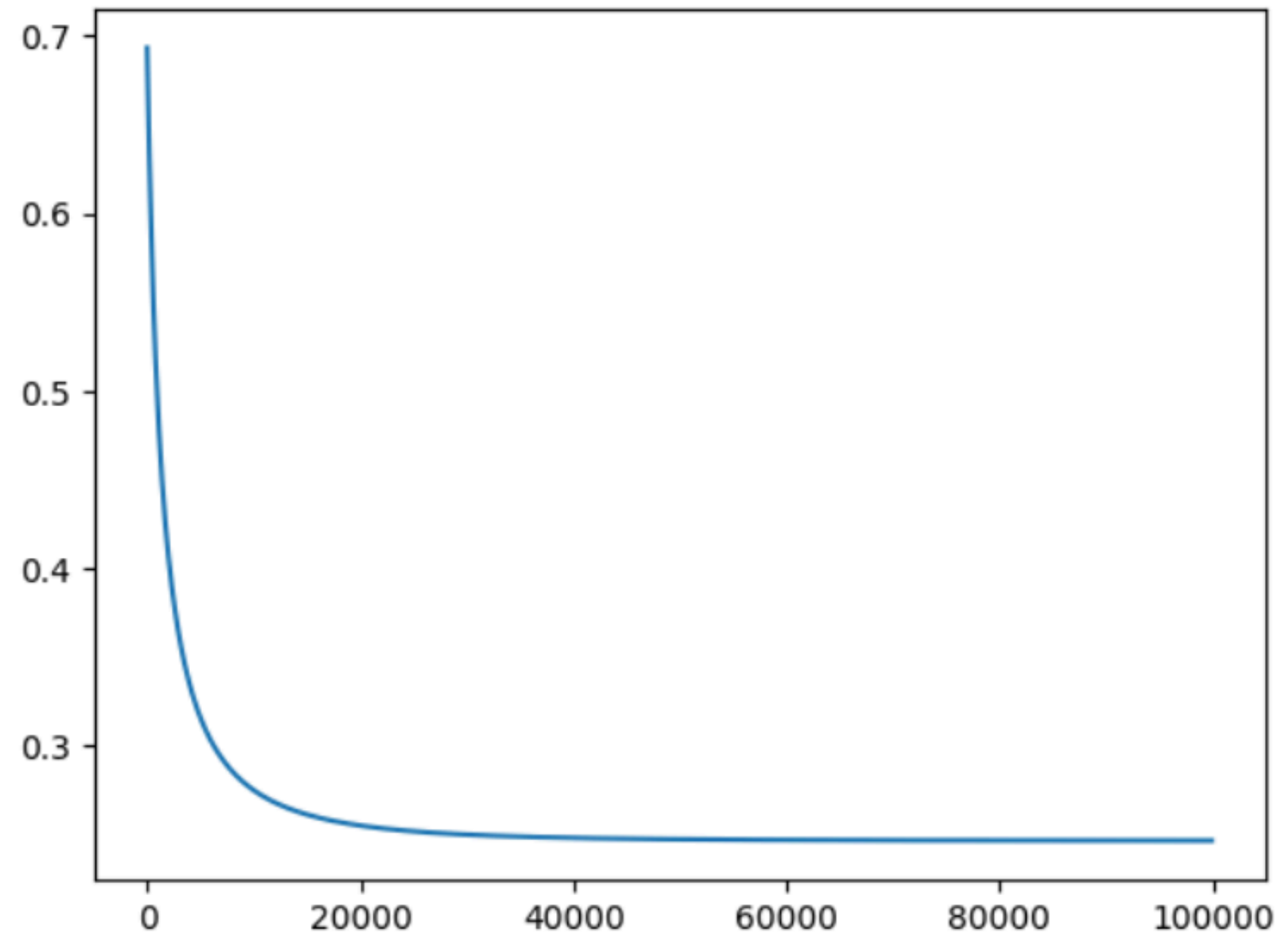
i tried logistic regression to
solve

Accuracy= 93.48125%

loss =1.5009976558721128

f1 score = 0.89368353

final cost = 0.246716174532

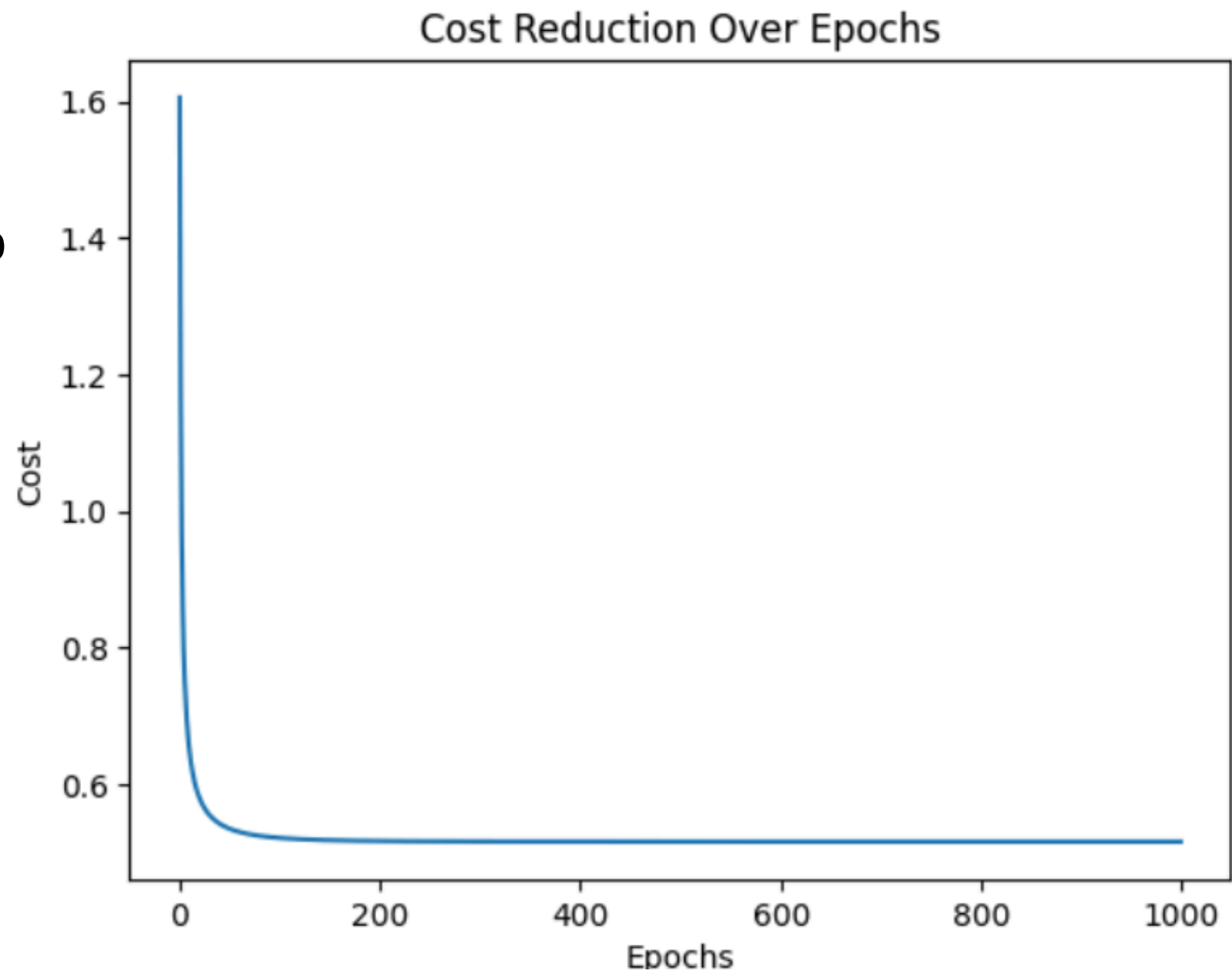


cost vs iteration graph

Multiple Classification

Training Accuracy: 83.45%

loss = 0.7487239394



Epoch 0, Cost: 1.6054
Epoch 50, Cost: 0.5344
Epoch 100, Cost: 0.5210
Epoch 150, Cost: 0.5177
Epoch 200, Cost: 0.5165
Epoch 250, Cost: 0.5161
Epoch 300, Cost: 0.5159
Epoch 350, Cost: 0.5158
Epoch 400, Cost: 0.5157
Epoch 450, Cost: 0.5157
Epoch 500, Cost: 0.5157
Epoch 550, Cost: 0.5157
Epoch 600, Cost: 0.5157
Epoch 650, Cost: 0.5157
Epoch 700, Cost: 0.5157
Epoch 750, Cost: 0.5157
Epoch 800, Cost: 0.5157
Epoch 850, Cost: 0.5157
Epoch 900, Cost: 0.5157
Epoch 950, Cost: 0.5157
Training Accuracy: 83.45%

Softmax Regression (Multinomial Logistic Regression)

KNN Method Multi_Classification

Loss: 3.6028

KNN Accuracy: 89.57%

This is the accuracy when I used 20% of train set as test set

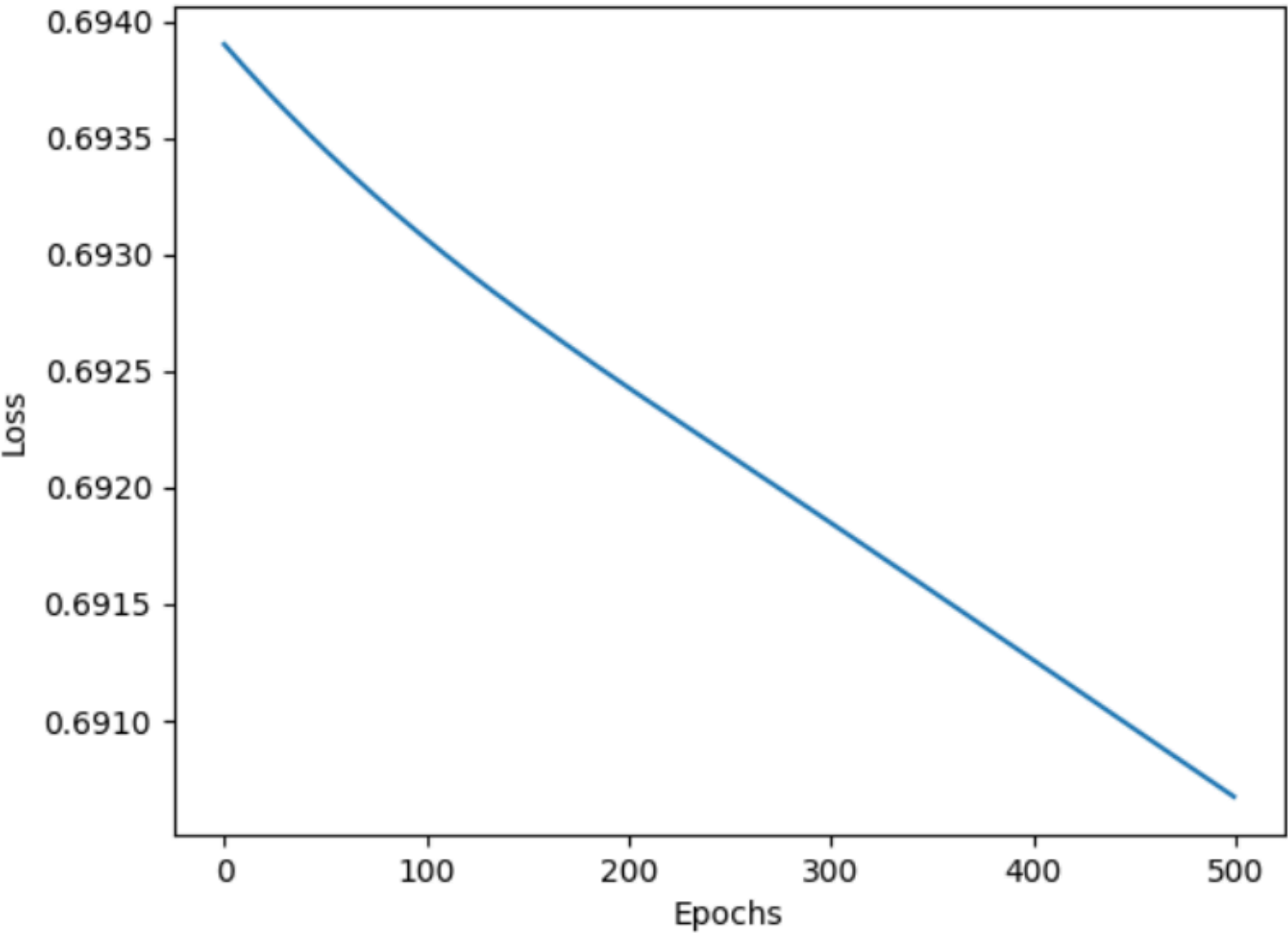
even on adding epoch the loss almost remained the same

neural network
binary

Hyperparameter Tuning as well as
Feedforward Neural Networks (FNN) and
Multilayer Perceptron (MLP) which is
advanced FNN

Epoch 0,	Loss: 0.6951
Epoch 50,	Loss: 0.6936
Epoch 100,	Loss: 0.6924
Epoch 150,	Loss: 0.6914
Epoch 200,	Loss: 0.6906
Epoch 250,	Loss: 0.6900
Epoch 300,	Loss: 0.6894
Epoch 350,	Loss: 0.6888
Epoch 400,	Loss: 0.6884
Epoch 450,	Loss: 0.6879

Loss Curve

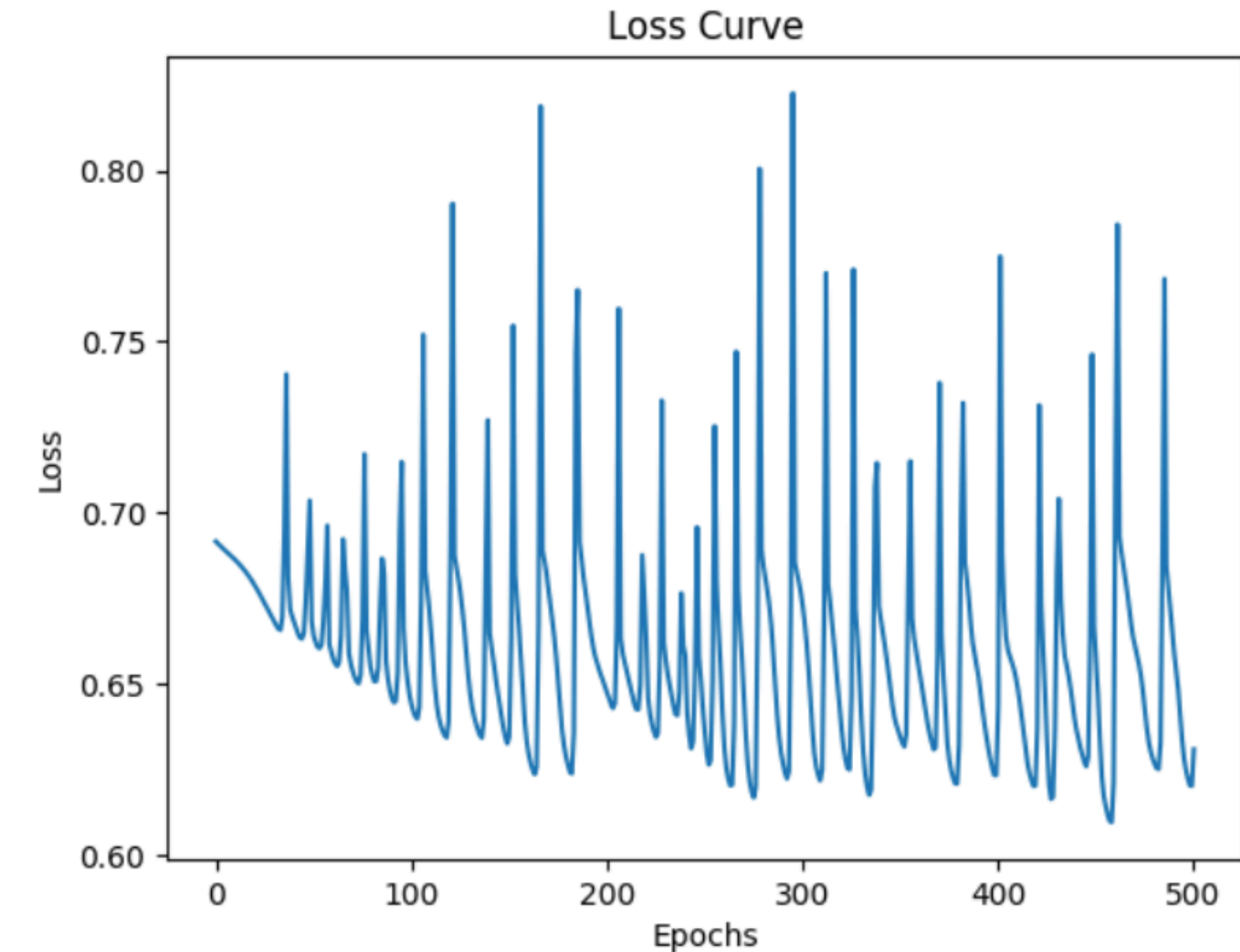


Final loss: 0.6869

Training Accuracy: 49.98%

Epoch 0, Loss: 0.6916
Epoch 50, Loss: 0.6640
Epoch 100, Loss: 0.6440
Epoch 150, Loss: 0.6345
Epoch 200, Loss: 0.6474
Epoch 250, Loss: 0.6357
Epoch 300, Loss: 0.6730
Epoch 350, Loss: 0.6343
Epoch 400, Loss: 0.6433
Epoch 450, Loss: 0.6573
Epoch 500, Loss: 0.6308

This is for learning rate
0.01

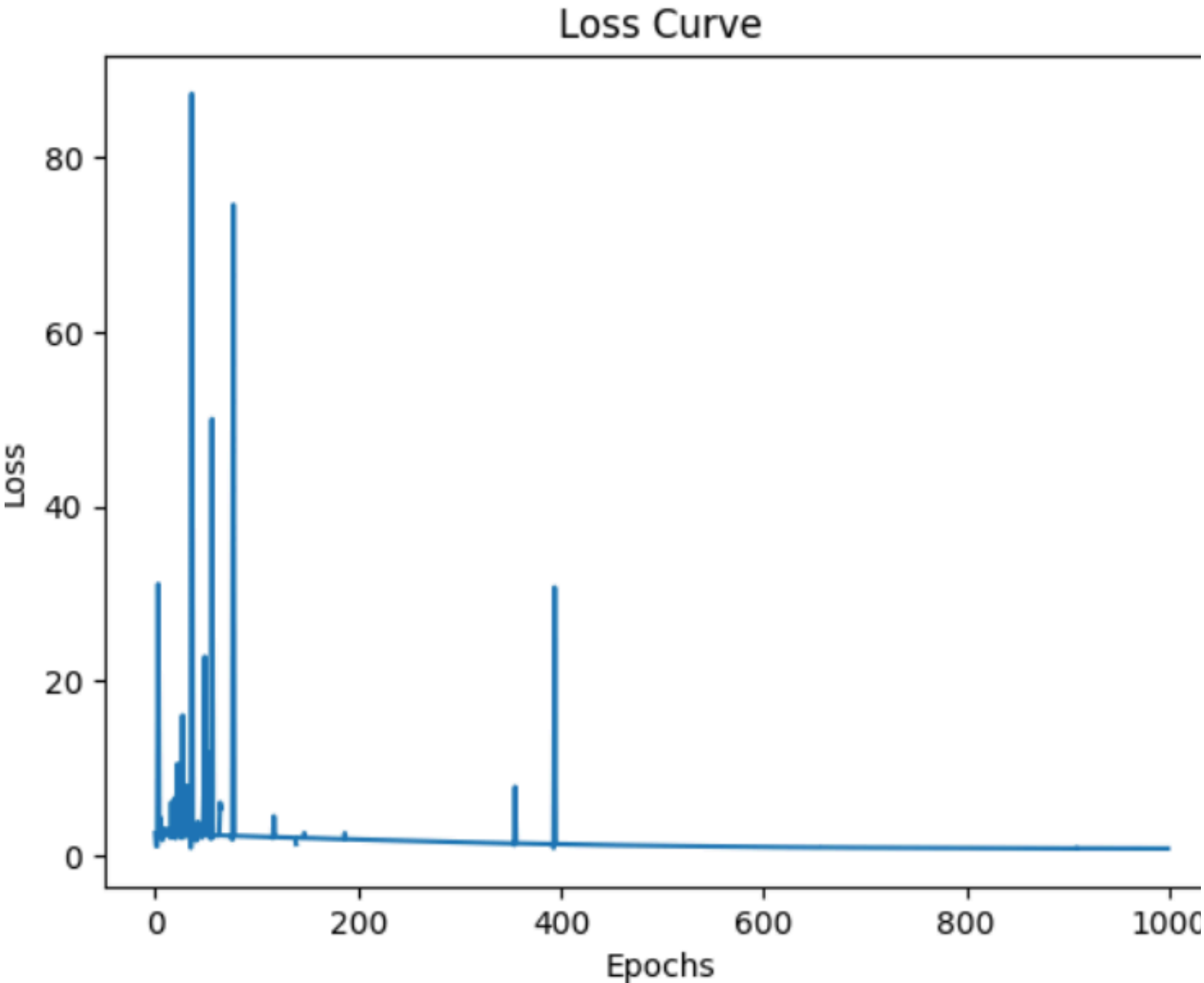


Training Accuracy: 59.02%

although the fluctuations are high but it gives better
accuracy in less time otherwise I will have to add
more epochs

neural network
class

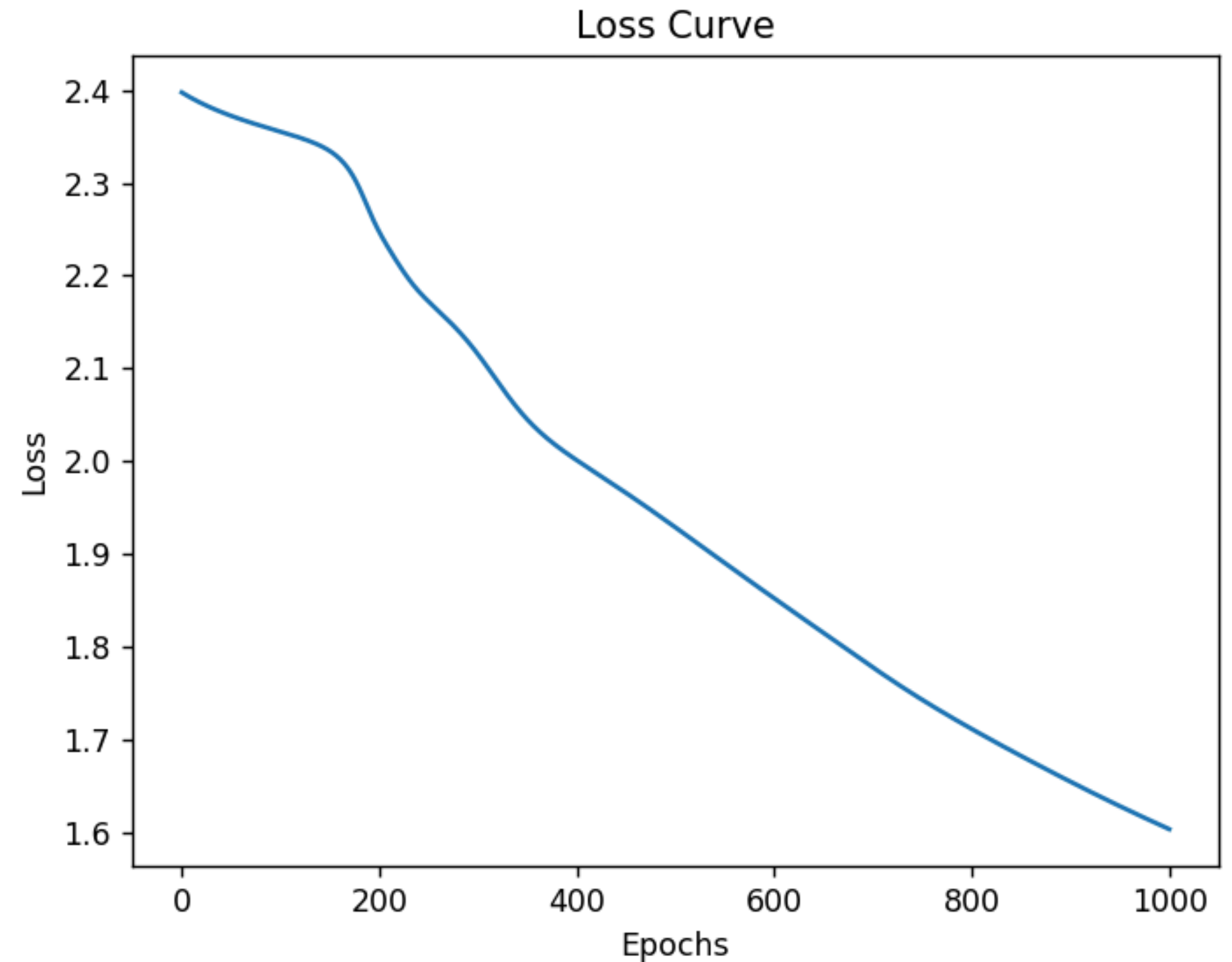
Hyperparameter Tuning as well as Feedforward
Neural Networks (FNN) and
Multilayer Perceptron (MLP) which is advanced FNN



Epoch 0, Loss: 2.3582
Epoch 100, Loss: 1.0917
Epoch 200, Loss: 0.8639
Epoch 300, Loss: 0.7971
Epoch 400, Loss: 0.7669
Epoch 500, Loss: 0.7501
Epoch 600, Loss: 0.7394
Epoch 700, Loss: 0.7320
Epoch 800, Loss: 0.7266
Epoch 900, Loss: 0.7225

Training Accuracy: 51.16%

Epoch 0, Loss: 2.3977
Epoch 100, Loss: 2.3554
Epoch 200, Loss: 2.2472
Epoch 300, Loss: 2.1151
Epoch 400, Loss: 2.0009
Epoch 500, Loss: 1.9284
Epoch 600, Loss: 1.8519
Epoch 700, Loss: 1.7773
Epoch 800, Loss: 1.7110
Epoch 900, Loss: 1.6541
Epoch 1000, Loss: 1.6029



this was with learning_rate = 0.07 in vs_code
but the Training Accuracy: 44.44%,decreased