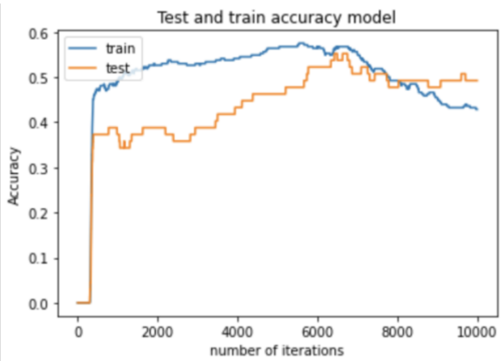
Perceptron Model with backward propogation

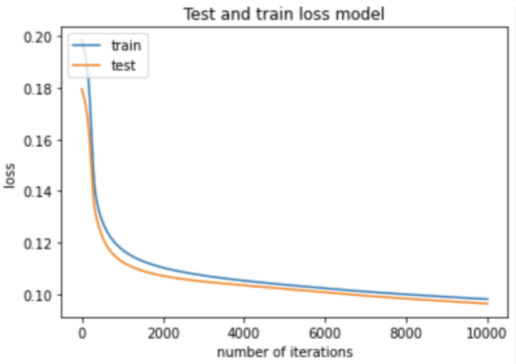
A Perceptron model was built from scratch, implementing forward and backward propagation. The penguins_iter.csv dataset was loaded into a data frame and preprocessed. The same preprocessing steps that were done in assignment 2 was done here.

We removed the missing values for a few columns, filled some of them with the column's mean and one-hot encoded was performed on the categorical features. The entire dataset was then normalized and the target variable that we tried to predict using perceptron model was "Sex" of penguins. We used sklearns library to split the dataset into train and test with 80% of the data set in train and the remaining in test.

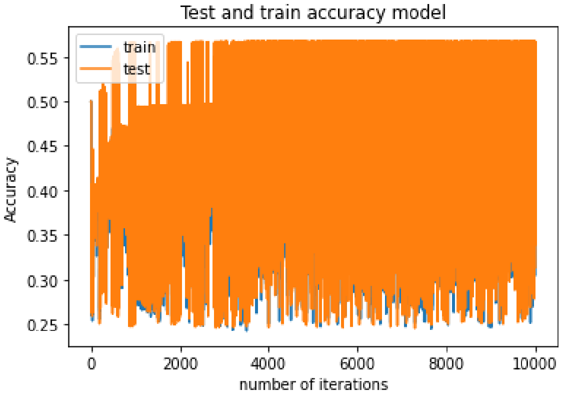
We decided on a learning rate of **0.001** and ran the model **10000** times. A perceptron model was built with an input layer with 12 nodes, one hidden layer with 266 nodes and the output layer, with 2 nodes. Two sets of weights were initialized along with the bias. We used sigmoid activation function to calculate the predicted y values and used the derivative of this function to perform back propagation. We then used this final model to predict our test data set to see how good our model has performed. However, when we compared this model to the logistic regression model (error rate of **0.116%**) that was performed in assignment 2, the perceptron model performed poorly. We created 2 different versions of a perceptron model with the same parameters to see if one performs better than the other.

The below graph shows the accuracy and mse plot for the first mode. We obtained a test accuracy of **49.25%**





The second model also performed poorly with test Mean Square Error **53.91%**. Below graph shows the train and test accuracy plot.


Test and train accuracy model
train
test
Accuracy
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0
2000
4000
6000
8000
10000
number of iterations