Machine Learning Report

Shivansh Shrivastava 2020A7PS2095H

Akul V Athreye 2020AAPS1768H

Rohit Reddy Palpunuri 2020AAPS1325H

Logistic Regression 1 (LE1)

As per the task requirement, we got a maximum accuracy of **56.4251207729469%** with a probability of **0.3**. We trained the model for 1000 epochs, with a **learning rate = 0.001** and **mini-batch gradient descent** was used with batch size 32. We dropped the null values from the data frame. All the accuracy percentages can be found in this excel sheet.

Logistic Regression 2 (LE2)

After filling null values with the mean and normalising the data, we got the best accuracy of **97.0212765957447%** with a standard deviation of 0.875821917392078. We had a **probability split of 0.5** for 1000 epochs and a **learning rate of 0.01**, and **mini-batch gradient descent** was used with batch size 32.
We tested Models LR1 and LR2 for in **total of 900 readings.** Ten random splits for Batch Gradient Descent, Mini-batch Gradient Descent and Stochastic Gradient Descent using learning rates 0.01, 0.001 and 0.0001. We varied the probability threshold from 0.5 to 0.3, 0.4, 0.6 and 0.7.
The graphs for Cost Function vs Iterations are built within the Colab notebook.

Records:
https://docs.google.com/spreadsheets/d/1P0jEENyu5uQrssUoyhcwQnRHhMwoUQsD/edit?usp=sharing&ouid=109813234497090782107&rtpof=true&sd=true

Observation: we can see we get 0 (majorly) as accuracy for LE1 due to smaller dataset(null vales dropped), and very low accuracies, due to no convergence(no normalisation). To make it work, we can increase the number of epochs by a factor of ten or even a hundred with increased learning rates.

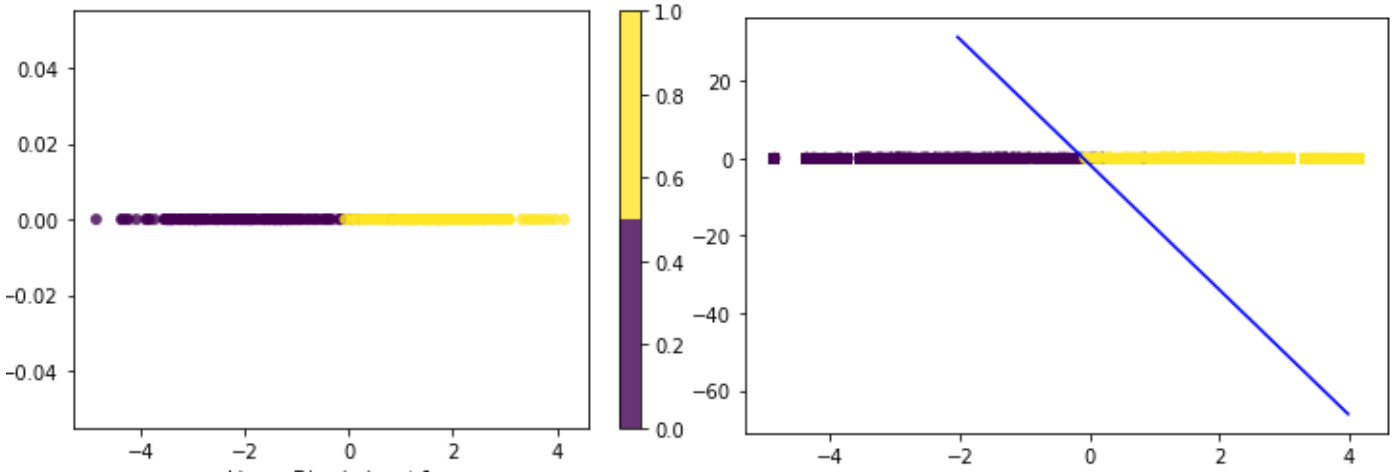
We can also see LE2 working best from the same data with a probability split of 0.5 as we have normalised data and cleaned up null fields.

Conclusion: From LE1 and LE2, LE2 performs way better than LE1 due to normalisation. It enables faster convergence.

Fischer's Linear Discriminant Analysis

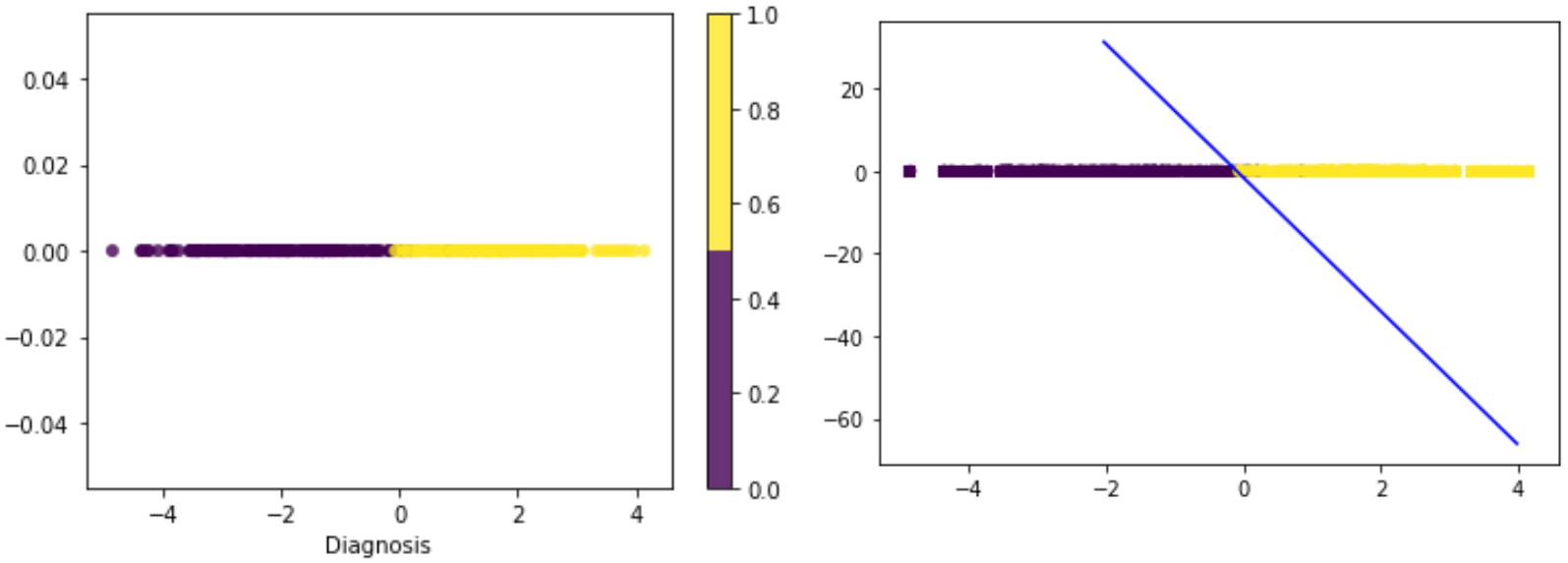
Learning Task - 1 (FLDM1)

After implementing FE1 and FE2, we got an accuracy score of 96.85%. Below scatter plot shows the Benign(yellow) and Malignant(purple) data points on a 1D, single-feature plane and the decision boundary.



Learning Task - 2 (FLDM2)

After shuffling the data set's features and undergoing the same process of transformation and fitting, we can get an **accuracy score of 96.85%** and the following to scatter plots.



Records: FLDA

Conclusion: The accuracy scores and decision boundary are relatively unchanged even after randomly shuffling the features' order. Thus the models can be considered to be **precisely the same.**

Perceptron Learning Algorithm

The model is developed based on the perceptron algorithm
On training the model with the given dataset. We obtained a mean testing **accuracy of 89.36679488% for the PM1 model.** Which means there is almost 90 percent chance that the data will be predicted correctly or we can say that there is 90 percent chance that the data in linearly separable by the decision boundary defined by the value expression

- ## Learning Task 1:

  Training the model by changing the ordering of the training samples resulted in a mean testing accuracy of **62.73294126%** of the PM2 model. PM1 is more accurate than PM2 by **26.63385362%**

- ## Learning Task 2:

  Normalising our dataset and then training the model resulted in a mean testing accuracy of **94.02086594%** of the PM3 model. The PM3 classifier, however, performs with higher accuracy compared to PM1 by **4.65407106%**

- Learning Task 3:

  The PM4 model developed by first rearranging the order of the features in the dataset and training the model with the dataset gave a testing accuracy of **89.36679488%**
  The PM4, and PM1 classifiers perform with the same accuracy on the testing data - rearranging the order of the feature does not affect the algorithm

Iteration	Split Factor	PM1	PM2	PM3	PM4
1	0.4	92.10526316	61.98830409	95.32163743	92.10526316
2	0.45	79.87220447	62.30031949	95.84664537	79.87220447
3	0.5	77.54385965	61.75438596	95.0877193	77.54385965
4	0.55	94.94163424	60.70038911	93.38521401	94.94163424
5	0.6	93.85964912	60.0877193	93.85964912	93.85964912
6	0.65	89	60.5	92.5	89
7	0.67	84.04255319	60.63829787	92.55319149	84.04255319
8	0.7	94.73684211	64.32748538	91.8128655	94.73684211
9	0.75	93.70629371	65.73426573	95.1048951	93.70629371
10	0.8	93.85964912	69.29824561	94.73684211	93.85964912
	MEAN ACCURACY	89.36679488	62.73294126	94.02086594	89.36679488
	Difference between model and PM1	0	26.63385362	-4.65407106	0

Comparative Study

The best possible model which can be implemented is the LE2 model, with the highest average accuracy score of 97.0212765957447% with the FLDM1 & FLDM2 models coming in a close second with an average accuracy score of **96.8503937007874%**. The perceptron algorithm, however, isn't highly promising, with the best accuracy of **84.0425531914%.**

One of the main reasons Logistic Regression may perform better than the Fisher Discriminant and Perceptron Algorithm is that **it can handle non-linearly separable data by using a non-linear transformation of the input variables.** This is accomplished by adding a non-linear function of the input variables to the linear equation, allowing the logistic regression model to capture complex relationships between the input and output variables.

In contrast, Fisher Discriminant and Perceptron Algorithms work only with linearly separable data, meaning they can only classify data separated by a straight line or

hyperplane. **If the data is not linearly separable, these algorithms may struggle to find a reasonable boundary between the classes, leading to poor performance.**

Another reason why **Logistic Regression may perform better than Fisher Discriminant and Perceptron Algorithm is because it can handle imbalanced datasets more effectively. Logistic Regression can adjust the weights given to different classes based on their relative frequencies in the data, improving its ability to classify the minority class correctly.**

In summary, while Fisher Discriminant and Perceptron Algorithm have their strengths, Logistic Regression is often preferred over them because it can handle non-linearly separable data and imbalanced datasets more effectively, leading to better performance in many real-world scenarios.